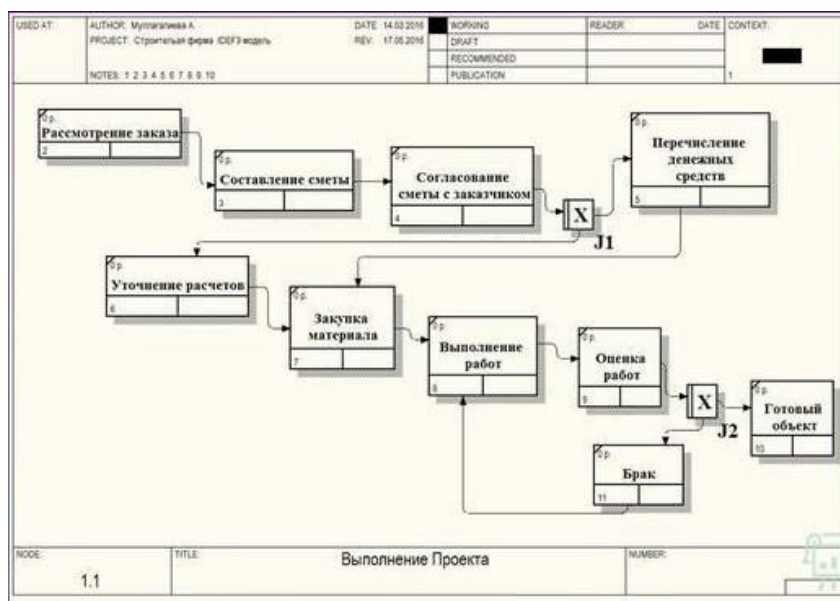


**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**  
**СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ**  
**ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ**  
**ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО**  
**БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»**  
**КАФЕДРА ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**



Ткачук Е.О.

## Проектирование сложных систем

**Методическое пособие**

**для проведения лабораторных работ и практических занятий**

Ростов-на-Дону

2019 г.

УДК 004.415.2

**Ткачук Е.О.** Проектирование сложных систем: Методическое пособие для проведения лабораторных работ и практических занятий. - Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019. – 164 с.: ил.

В пособии даются организационно-методические указания к лабораторным вопросам и практическим занятиям, порядок их выполнения и оформления отчётов.

Предназначено для студентов всех специальностей обеих форм обучения, изучающих дисциплину «Проектирование сложных систем», а также может быть полезно всем остальным студентам, желающим самостоятельно изучать современные методы проектирования программных продуктов.

**Составители:**

доцент кафедры ИВТ Ткачук Е.О.

**Рецензент:** доц. кафедры ИВТ СКФ МТУСИ, к.т.н. доц. А.Н. Чикалов.

Издание рассмотрено и утверждено

на заседании кафедры ИВТ

26.08.2019 года (протокол № 1)

© СКФ МТУСИ, 2019

© Ткачук Е.О., 2019 г.

|   |            |
|---|------------|
| <b>1. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1. ОПИСАНИЕ КОМПАНИИ.....</b>  | <b>7</b>   |
| 1.1. Краткая информация о компании "МЕД" .....  | 7          |
| 1.2. Отчет об обследовании .....  | 9          |
| 1.3. Разработка моделей бизнес-процессов предприятия оптовой торговли лекарствами .....                                 | 14         |
| <b>2. ЛАБОРАТОРНАЯ РАБОТА 1 ФОРМИРОВАНИЕ ДИАГРАММ И СПИСКОВ БИЗНЕС-ПРОЦЕССОВ. ....</b>                                  | <b>16</b>  |
| 2.1. Задание 1. Формирование физической диаграммы .....   | 16         |
| 2.2. Задание 2. Формирование списка бизнес-процессов .....  | 19         |
| 2.3. Задание 3. Построение диаграммы действий .....   | 21         |
| <b>3. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2. ПРЕДВАРИТЕЛЬНОЕ ФОРМИРОВАНИЕ ТАБЛИЦ .....</b>   | <b>28</b>  |
| 3.1. Задание 4. Формирование таблицы операций.....  | 28         |
| 3.2. Задание 5. Формирование таблицы описания документов .....  | 32         |
| 3.3. Задание 6. Построение диаграммы действий.....  | 36         |
| 3.4. Задание 7. Формирование таблицы операций.....  | 41         |
| 3.5. Задание 8. Формирование таблицы описания документов .....  | 44         |
| 3.6. Задание 9. Построение диаграммы действий .....   | 47         |
| <b>4. ЛАБОРАТОРНАЯ РАБОТА 2. ФОРМИРОВАНИЕ ТАБЛИЦЫ ОПЕРАЦИЙ.....</b>   | <b>53</b>  |
| 4.1. Задание 10. Формирование таблицы операций.....   | 53         |
| 4.2. Задание 11. Формирование таблицы описания документов .....   | 56         |
| 4.3. Задание 12. Построение диаграммы действий .....  | 59         |
| 4.4. Задание 13. Формирование таблицы операций.....   | 59         |
| 4.5. Задание 14. Формирование таблицы описания документов .....   | 59         |
| <b>5. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3. ПОСТРОЕНИЕ ДИАГРАММЫ ДЕЙСТВИЙ. ....</b>   | <b>61</b>  |
| 5.1. Задание 15. Построение диаграммы действий .....  | 61         |
| 5.2. Задание 16. Формирование таблицы операций.....   | 61         |
| 5.3. Задание 17. Формирование таблицы описания документов .....   | 61         |
| <b>6. ЛАБОРАТОРНАЯ РАБОТА 3. ПРОЕКТИРОВАНИЕ РЕАЛИЗАЦИИ ОПЕРАЦИЙ БИЗНЕС-ПРОЦЕССА В ИНФОРМАЦИОННОЙ СИСТЕМЕ (ИС) .....</b> | <b>64</b>  |
| 6.1. Задание 18. Проектирование реализации операций бизнес-процесса в информационной системе (ИС) .                     | 64         |
| 6.2. Задание 19. Проектирование реализации операций бизнес-процесса в информационной системе (ИС) .                     | 67         |
| 6.3. Задание 20. Проектирование реализации операций бизнес-процесса в информационной системе (ИС)                       | 68         |
| 6.4. Задание 21. Проектирование реализации операций бизнес-процесса в информационной системе .....                      | 70         |
| <b>7. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4. РАЗРАБОТКА ГЛОССАРИЯ .....</b>  | <b>72</b>  |
| <b>8. ЛАБОРАТОРНАЯ РАБОТА 4. ПОДРОБНОЕ ОПИСАНИЕ КЛЮЧЕВЫХ УС .....</b>   | <b>75</b>  |
| <b>9. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5. ОСНОВЫ РАБОТЫ С CASE СРЕДСТВОМ ALLFUSION PROCESS MODELER .....</b>                        | <b>79</b>  |
| <b>10. ЛАБОРАТОРНАЯ РАБОТА 5 РАБОТА С CASE СРЕДСТВОМ ALLFUSION PROCESS MODELER .....</b>                                | <b>94</b>  |
| <b>11. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6. ОСНОВЫ РАБОТЫ С CASE СРЕДСТВОМ ALLFUSION PROCESS MODELER</b>                             | <b>105</b> |

|   |     |
|---|-----|
| 12. ЛАБОРАТОРНАЯ РАБОТА 6. ИСПОЛЬЗОВАНИЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT СОВМЕСТНО С САЙТОМ GITHUB.....   | 109 |
| 13. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 7. ИСПОЛЬЗОВАНИЕ RAD ТЕХНОЛОГИИ ДЛЯ ПРОТОТИПИРОВАНИЯ.....                    | 122 |
| 14. ЛАБОРАТОРНАЯ РАБОТА 7. СОЗДАНИЕ ПРОТОТИПА ИС С ИСПОЛЬЗОВАНИЕМ RAD ТЕХНОЛОГИЙ.....                 | 140 |
| 15. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 8. СИНТЕЗ ВНУТРИ МАШИННОГО ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ СЛОЖНОЙ СИСТЕМЫ ..... | 157 |
| 16. ЛАБОРАТОРНАЯ РАБОТА 8. ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СХЕМЫ ДАННЫХ.....                                   | 162 |

## **ТЕМЫ индивидуальных заданий**

1. Разработка АРМ для аудитора
2. Разработка АРМ для учета движения готовой продукции на предприятии
3. Разработка АРМ для управления запасами на производстве
4. Разработка АРМ для автоматизации документооборота на кафедре ВУЗ
5. Разработка АРМ для оценщика автомобилей и дорогостоящего оборудования
6. Разработка АРМ для ведения расписания учебных занятий
7. Разработка АРМ туроператора для туристического агентства
8. Разработка АРМ для учета и анализа финансовых результатов на производстве
9. Разработка АРМ для учета основных средств на производстве
10. Разработка АРМ для учета движения продукции на аптечном складе
11. Разработка АРМ для библиотеки ВУЗа
12. Разработка АРМ для учета расчетов по долгам предприятия
13. Разработка АРМ для учета и анализа затрат на выпуск и реализацию готовой продукции на производстве
14. Разработка АРМ для учета движения продукции на оптовом складе
15. Разработка АРМ для учета товаров в мелкооптовой торговой фирме
16. Разработка АРМ для статистических расчетов
17. Разработка АРМ для управления проектами
18. Разработка АРМ для учета денежных средств на производстве

## 1. Практическое занятие 1. Описание компании

На основе предварительной информации сформировано и согласовано с заказчиком общее представление о проекте:

**Видение выполнения проекта и границы проекта** - документ, который кратко описывает, в каких подразделениях и в какой функциональности будет внедряться ИС. Затем выполняется детальное обследование предприятия, результаты которого оформляются в виде отдельного документа - отчета об обследовании.

Отчет об обследовании содержит следующие разделы:

- Анализ существующего уровня автоматизации.
- Составляется список программного обеспечения, используемого в компании, и приводятся данные об использовании этих пакетов в каждом из подразделений организации.
- Общие требования к ИС
- Формулируются общие требования к функциональности разрабатываемой системы.
- Формы документов
- Устанавливается перечень и структура документов, которые должны формироваться системой.
- Описание системы учета
- Описание системы учета включает в себя следующие документы:
  - Учетная политика компании
  - План счетов и используемых аналитик
  - Список типовых хозяйственных операций и их отражение в проводках
  - Описание справочников
  - По каждому справочнику, проектируемому в системе, дается описание необходимой иерархической структуры.
- Организационная диаграмма
- Организационная диаграмма используется для отражения организационной структуры подразделений предприятия и их зон ответственности.
- Описание состава автоматизируемых бизнес-процессов

- Все бизнес-процессы компании должны быть перечислены в общем списке и каждый должен иметь свой уникальный номер.
- Диаграммы прецедентов
- Для выделения автоматизируемых бизнес-процессов и их основных исполнителей используются диаграммы прецедентов.
- Физическая диаграмма
- Физическая диаграмма служит для того, чтобы описать взаимодействие организации на верхнем уровне с внешними контрагентами.
- Описания бизнес-процессов (книга бизнес-процессов).

Далее в отчет об обследовании включается книга бизнес-процессов, содержащая подробное описание автоматизируемых бизнес-процессов. Модели бизнес-процессов позволяют выделить отдельные операции, выполнение которых должно поддерживаться разрабатываемой ИС. На последнем этапе осуществляется отображение модели предметной области на функциональность типовой системы - выбираются модули системы для поддержки выделенных операций, определяются особенности их настройки, выявляется необходимость разработки дополнительных программных элементов.

### 1.1. Краткая информация о компании "МЕД"

Компания - дистрибьютор "МЕД" закупает медицинские препараты отечественных и зарубежных производителей и реализует их через собственную дистрибьюторскую сеть и сеть аптек. Компания осуществляет доставку товаров как собственным транспортом, так и с помощью услуг сторонних организаций. Основные бизнес-процессы компании - закупки, складирование запасов, продажи, взаиморасчеты с поставщиками и клиентами. Уровень конкуренции для компании в последнее время возрос, так как на рынок вышли два новых конкурента, к которым перешла часть клиентов и ряд наиболее квалифицированных сотрудников ЗАО "МЕД". ЗАО "МЕД" имеет два филиала - в Курске и Санкт-Петербурге. Каждый филиал функционирует как самостоятельное юридическое лицо, являясь полностью принадлежащей ЗАО "МЕД" дочерней компанией. По предварительным планам, Компания намерена открыть также дочернее предприятие для организации производства в непосредственной близости к своим заказчикам.

### Адреса и телефоны

Москва, К-123 Центральная улица, д. 20, стр. 7, офис 709

Телефон: (095) 345-6789, факс: (095) 345-9876

### **Контактные лица**

Борис Нефедьев - Генеральный директор

Дмитрий Кононов - Исполнительный директор

Артур Иванченко - Директор по маркетингу

### **Сотрудники**

На момент проведения Диагностики штат компании составляет 110 сотрудников.

Основными целями проекта автоматизации компании "МЕД" являются:

- Разработка и внедрение комплексной автоматизированной системы поддержки логистических процессов компании.
- Повышение эффективности работы всех подразделений компании и обеспечение ведения учета в единой информационной системе.

### ***Видение выполнения проекта и границы проекта***

В рамках проекта развертывание новой системы предполагается осуществить только в следующих подразделениях ЗАО "МЕД":

- Отдел закупок;
- Отдел приемки;
- Отдел продаж;
- Отдел маркетинга;
- Группа планирования и маркетинга;
- Группа логистики;
- Учетно-операционный отдел;
- Учетный отдел;
- Отдел сертификации (в части учета сертификатов на медикаменты);
- Бухгалтерия (только в части учета закупок, продаж, поступлений и платежей).



Не рассматривается в границах проекта автоматизация учета основных средств, расчета и начисления заработной платы, управления кадрами. Выходит за рамки проекта автоматизация процессов взаимоотношений с клиентами.

## 1.2. Отчет об обследовании

Список программного обеспечения, используемого компанией на момент обследования

1. "1С: Предприятие 7.7" ("Бухгалтерия", "Торговля", "Зарплата", "Кадры", "Касса", "Банк") для работы бухгалтерии.
2. Две собственные разработки на базе конфигуратора "1С" - "Закупки" и "Продажи".
3. Собственная разработка на базе FOXPRO для финансового отдела.
4. Excel для планирования продаж.

### *Существующий уровень автоматизации*

|  |  |
|--|--|
| Количество рабочих станций, всего:                               | 90   |
| Количество сотрудников отдела IT                                 | 2  |
| Количество ПК, одновременно работающих в сети                    | 50   |
| Наличие и форма связи с удаленными объектами                     | Терминальная связь со складом  |
| Количество рабочих станций на удаленном объекте                  | на 8   |
| Характеристики компьютеров                                       | От Celeron 600 и выше  |
| Операционная система   | Windows 98, XP   |
| Системы, которые представляется возможным оставить без изменения | "1С: Предприятие 7.7" в модульном составе "Бухгалтерия", "Зарплата", "Кадры", для работы бухгалтерии |

### *Общие требования к информационной системе*

Одно из основных требований компании "МЕД" к будущему решению состоит в том, чтобы оно было построено на фундаменте единой интегрированной системы, а работа всех сотрудников велась в одном информационном пространстве.

Ключевые функциональные требования к информационной системе:

- Мощные средства защиты данных от несанкционированного доступа.

Разграничения доступа к данным в соответствии с должностными обязанностями.

- Возможность удаленного доступа.
- Управление запасами. Оперативное получение информации об остатках на складе.
- Управление закупками. Планирование закупок в разрезе поставщиков.
- Управление продажами. Контроль лимита задолженности с возможностью блокировки формирования отгрузочных документов.
- Полный контроль взаиморасчетов с поставщиками и клиентами.
- Получение управленческих отчетов в необходимых аналитических срезах - как детальных для менеджеров, так и агрегированных для руководителей подразделений и директоров фирмы.

### Примеры форм отчетных документов

| Отчет о дебиторской задолженности                              |                                  |                      |                           |                                |  |                                   |                            |                 |
|--|----------------------------------|----------------------|---------------------------|--------------------------------|--|-----------------------------------|----------------------------|-----------------|
| Регистрационн<br>ый номер                                      | Клие<br>нт                       | Догово<br>р          | Дата<br>договора          | Сумма по<br>договору           | Сумма<br>задолженнос<br>ти             | Ожидаемый<br>срок платежа         | Комментар<br>ий            |                 |
|  |                                  |                      |                           |                                |  |                                   |                            |                 |
| Итого  |                                  |                      |                           |                                |  |                                   |                            |                 |
| Отчет о кредиторской задолженности                             |                                  |                      |                           |                                |  |                                   |                            |                 |
| Информация о<br>материалах/комплектую<br>щих, услугах, работах |                                  | Поставщ<br>ик        | №<br>договор<br>а         | Сумма<br>по<br>договор<br>у    | Срок<br>оплаты<br>по<br>договор<br>у   | Дата<br>оплат<br>ы                | Сумма<br>задолженнос<br>ти | Комментар<br>ий |
|  |                                  |                      |                           |                                |  |                                   |                            |                 |
| Отчет о требуемых закупках                                     |                                  |                      |                           |                                |  |                                   |                            |                 |
| Инвент<br>арный код  | Название<br>материала/товар<br>а | Ед.<br>измерени<br>я | Тре<br>буется<br>закупить | Предыдущая дата приобретения   |  |                                   |                            |                 |
|  |                                  |                      |                           | Назв<br>ание<br>поставщик<br>а | Дата<br>последнего<br>приобретени<br>я | Стоимо<br>сть<br>приобретени<br>я |                            |                 |
|  |                                  |                      |                           |                                |  |                                   |                            |                 |

### Описание системы учета

ЗАО "МЕД" использует типовой российский план счетов, три аналитики (контрагенты, договора, регионы).

### Фрагмент плана счетов компании

| Номер<br>бухг. счета | Наименование счета                        |
|----------------------|---|
| 01.000               | Основные средства                         |
| 02.000               | Амортизация основных средств              |
| 03.000               | Доходные вложения в материальные ценности |
| 04.000               | Нематериальные активы                     |
| 05.000               | Амортизация нематериальных активов        |
| 08.000               | Вложения во внеоборотные активы           |
| 10.000               | Материалы                                 |
| 10.100               | Сырье и материалы                         |
| 10.200               | Прочие материалы                          |
| 10.300               | Инвентарь и хозяйственные принадлежности  |
| 14.000               | Резервы под снижение стоимости МЦ         |
| 16.000               | Отклонение в стоимости МЦ                 |
| 19.000               | НДС по приобретениям                      |
| ...                  | ...                                       |

**Фрагмент учетной политики: Выручка и прибыль.** Выручка от реализации продукции и оказания услуг определяется по мере отгрузки реализованной продукции, оказания услуг и отражается в финансовой отчетности по методу начисления.

**Запасы.** Компания с целью определения фактической себестоимости товаров, реализованных в отчетном периоде, использует вариант их оценки по себестоимости первых по времени приобретения материалов (ФИФО).

### Описание справочников

Фрагмент описания справочников, используемых для автоматизации компании "МЕД", приведен в таблице.

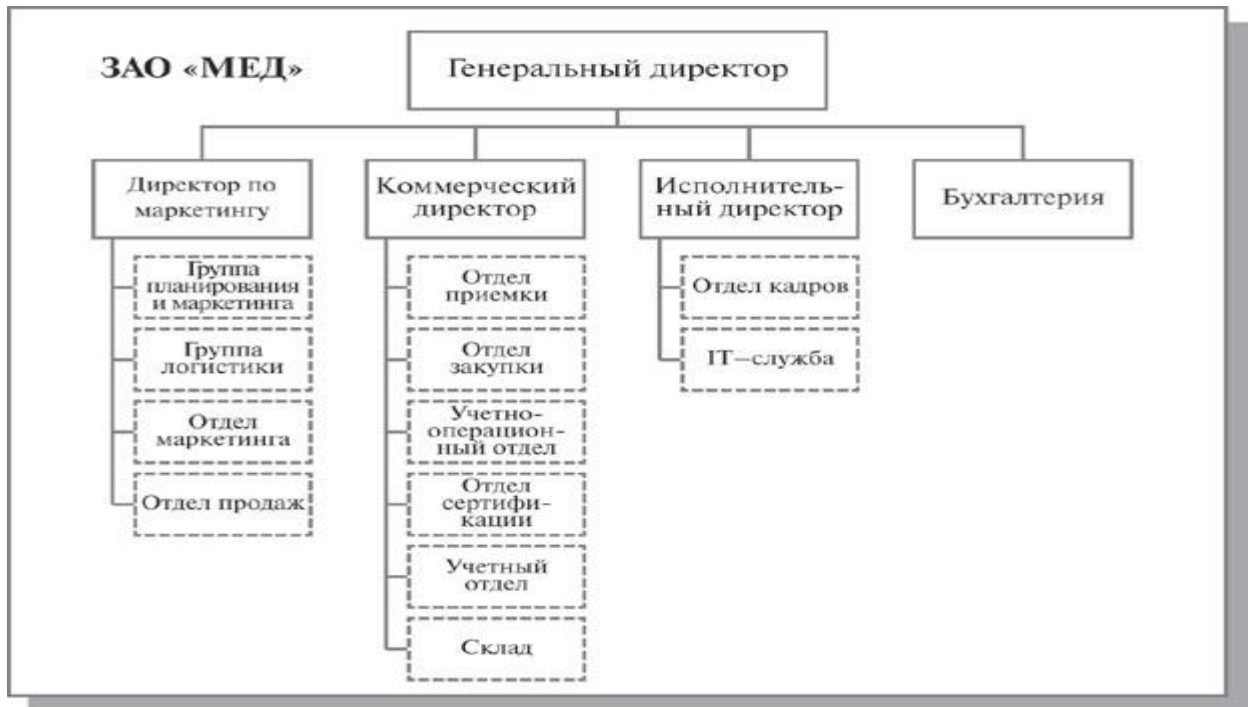
| № | Наименование справочника | Код           | Наименование                          |
|---|--------------------------|---------------|---------------------------------------|
|   | Клиенты                  | AC_Ap_00001   | Покупатель_АПТЕКИ                     |
|   |                          | AC_Ds_00001   | Покупатель_Дистрибьютеры              |
|   | Поставщики/подрядчики    | OTHER_00001   | Прочие                                |
|   |                          | B_00001       | Банки                                 |
|   |                          | L_00001       | Частные лица                          |
|   |                          | I_0001        | Страховые организации                 |
|   |                          | OTHER_00001   | Прочие                                |
|   | Договора                 |               |                                       |
|   | 1- наши услуги           | 1_COM_D/M/E   | Договор комиссии_Д/М/Г                |
|   |                          | 1_SERV_D_/M/E | Договор на оказание наших услуг_Д/М/Г |
|   | 2 - услуги нам           | 2_COM_D/M/E   | Договор комиссии_Д/М/Г по услугам нам |
|   |                          | 2_SERV_D/M/E  | Договор на указание Г услуг нам _Д/М  |
|   |                          | 2_COM_D/M/E   | Договор комиссии_Д/М/Г по услугам нам |

Код справочника отражает уровни иерархии. Справочники клиентов и договоров имеют трехуровневую структуру. Справочник поставщиков - двухуровневую структуру. В коде справочника для отображения уровня применен символ подчеркивания. Например, в коде справочника клиентов первый уровень обозначен символами "АС"-покупатель; второй

уровень - "Ap"-аптеки, "Ds"- дистрибьюторы; для обозначения третьего уровня предусмотрены пятизначные порядковые номера 00001, 00002 и т.д.

### Организационная диаграмма

Оргструктура предприятия оптовой торговли ЗАО "МЕД" имеет следующий вид:



### Описание состава автоматизируемых бизнес-процессов

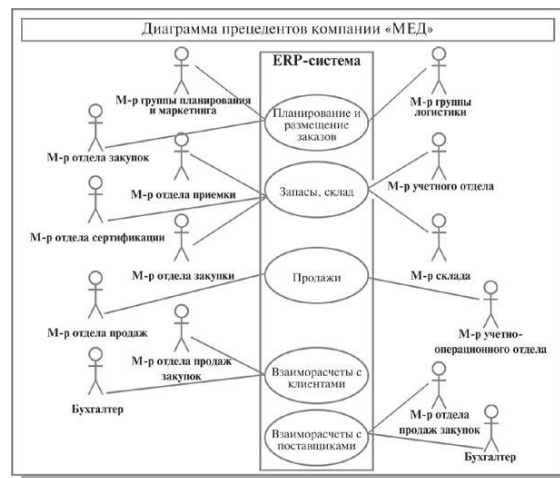
Бизнес-процессы компании, подлежащие автоматизации, приведены в следующей таблице:

| № п.п | Код бизнес-процесса | Наименование бизнес-процесса             |
|-------|---------------------|--|
| 1.    | Закуп-1             | Закупки                                  |
| 2.    | Склад-2             | Запасы-Склад                             |
| 3.    | Прод-3              | Продажи                                  |
| 4.    | Врасч-4             | Взаиморасчеты с поставщиками и клиентами |

Каждый бизнес-процесс имеет свой уникальный номер. Нумерация бизнес-процессов построена по следующему принципу: "префикс-номер", где префикс обозначает группу описываемых бизнес-процессов, а номер - порядковый номер бизнес-процесса в списке.

**Диаграмма прецедентов компании****"МЕД"**

На Диаграмме прецедентов представлены автоматизируемые бизнес-процессы компании и их исполнители.



### 1.3. Разработка моделей бизнес-процессов предприятия оптовой торговли лекарствами

**Термины**

**Внешняя статистика продаж** - статистика по продажам, получаемая из сети аптек;

**Внутренняя статистика продаж** - статистика по продажам, получаемая из отчетов продаж клиентам компании;

**Номенклатурная единица** - наименование медикамента, завода-изготовителя;

**ABC** - классификация товара по выручке от продаж клиентам;

**XYZ** - классификация товара по рейтингу популярности;

**Учетная цена** - цена товара у поставщика с учетом скидок;

**Действующие контракты** - контракты, по которым имеются обязательства сторон на определенный период времени;

**График поставок** - очередность обращения к поставщикам, необходимая для поддержания деловых отношений;

**Страховой запас** - минимальный запас товара, необходимый для покрытия потребностей до момента поставки новой партии товара.

Разработка информационных систем включает в себя несколько этапов. Однако всегда начальным этапом создания системы является изучение, анализ и моделирование деятельности заказчика.

Для того чтобы описать взаимодействие компании на верхнем уровне с внешними контрагентами, составляется физическая диаграмма. Для составления физической диаграммы в ходе первого интервью необходимо выяснить, кто является внешними контрагентами и какие у них основные функции.

## 2. Лабораторная работа 1      Формирование диаграмм и списков бизнес-процессов.

### 2.1. Задание 1. Формирование физической диаграммы

Составьте физическую диаграмму в соответствии с описанием деятельности компании дистрибьютора МЕД.

Компания дистрибьютор "МЕД" закупает медицинские препараты отечественных и зарубежных производителей и реализует их через собственную дистрибьюторскую сеть и сеть аптек. Планирование закупок компания осуществляет на основании статистики продаж, которую предоставляют сеть аптек и дистрибьюторы. Компания осуществляет доставку медикаментов как собственным транспортом, так и с помощью услуг сторонних организаций. Компания имеет собственный склад для хранения медикаментов.

#### ***Выполнение задания 1***

Компания осуществляет закупки у отечественных и зарубежных производителей, следовательно, контрагентами компании являются отечественные и зарубежные поставщики медикаментов. Компания пользуется услугами транспортных компаний для доставки медикаментов. Следовательно, транспортные компании являются внешними контрагентами. Кроме того, компания реализует медикаменты через дистрибьюторскую сеть и сеть аптек. Следовательно, контрагентами компании являются покупатели (дистрибьюторы, аптеки). Таким образом, внешними контрагентами компании "МЕД" являются поставщики (отечественные, зарубежные), покупатели (дистрибьюторы, аптеки), транспортные компании.

На физической диаграмме компания изображается прямоугольником, для отображения контрагентов используются графический символ Actor (фигурка человечка). Для изображения связей между компанией и контрагентами используются линии (Communications). Взаимодействия компании и внешних контрагентов должны быть поименованы, чтобы были понятны функции контрагентов по отношению к компании при знакомстве с физической диаграммой.

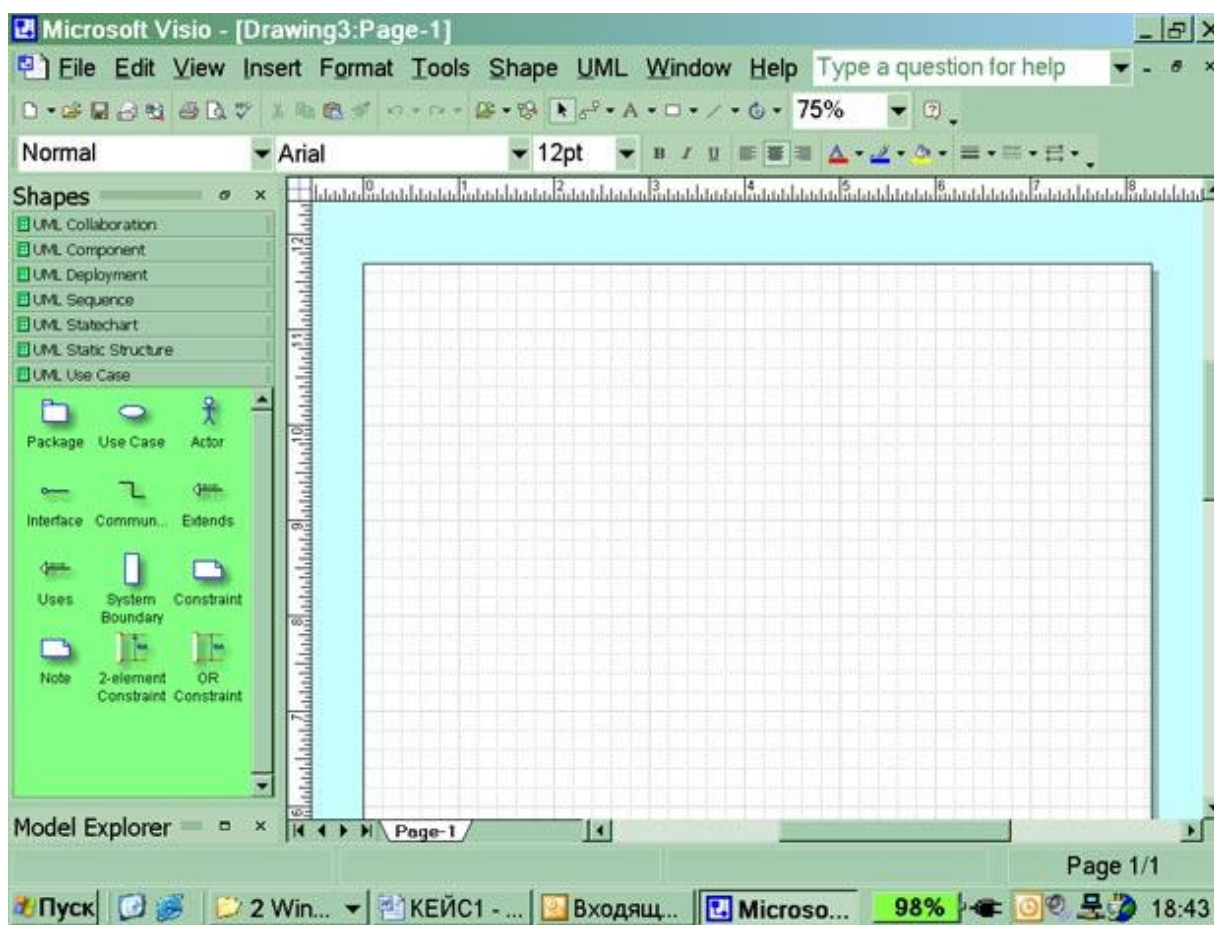


## Создание физической диаграммы в MS Visio:

Запустите MS Visio. (Кнопка "Пуск"/ "Программы" / MS Visio).


Появится окно, в котором необходимо выбрать папку **Software/ UML Model Diagram**.

В открывшемся списке форм (Shapes) для построения физической диаграммы следует выбрать пункт **UML Use Case**. В результате проделанных действий на экране появится окно, в левой части которого будет отображен набор графических символов, а в правой части - лист для рисования диаграммы (рис.1).



**Рис. 1.** Общий вид окна MS Visio

Для изображения прямоугольника на панели инструментов "Стандартная" найдите и зафиксируйте щелчком мыши пиктограмму с изображением прямоугольника. Затем, при нажатой правой клавиши мыши вы сможете нарисовать произвольного размера прямоугольник.

Для изображения на диаграмме контрагентов следует воспользоваться графическим символом с изображением человечка . Графический символ переносится на рабочее поле мышкой при нажатой правой клавише.

**Примечание.** Для последующего перемещения графических символов по рабочему полю необходимо зафиксировать пиктограмму Pointer Tool с изображением стрелки, размещенную на панели инструментов "Форматирование". Только после этого графический символ будет доступен для перемещения его мышкой.

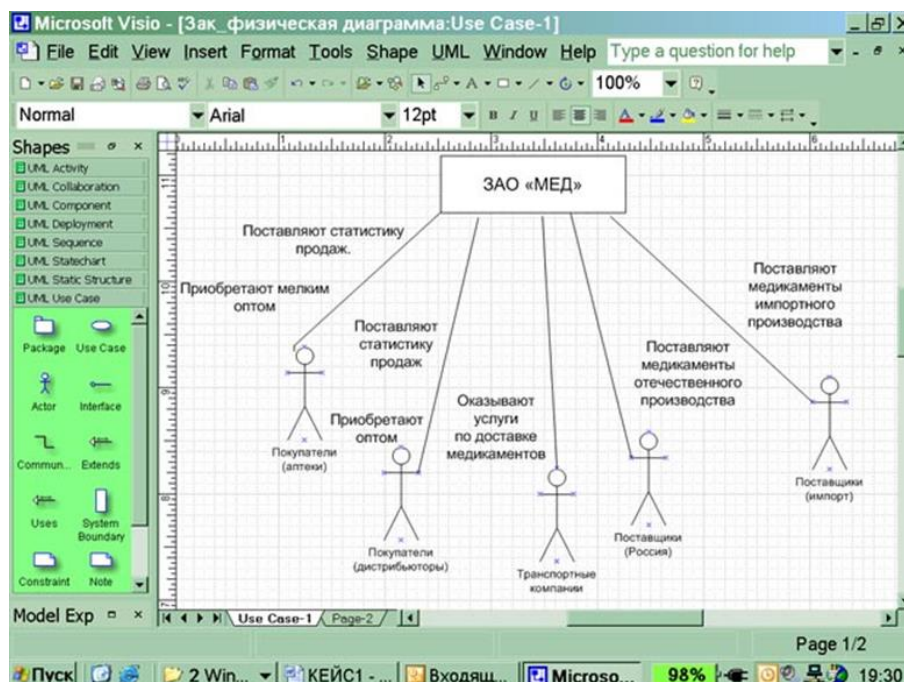
Соедините линиями изображение каждого контрагента с прямоугольником. Для этого на панели инструментов "Стандартная" щелчком мыши зафиксируйте пиктограмму с изображением линии Line Tool и при нажатой левой клавише мышки осуществите соединение фигур.

Внесите наименования контрагентов "Покупатели (аптеки)", "Покупатели (дистрибьюторы)", "Поставщики (Россия)", "Поставщики (импорт)", "Транспортные компании". Для того чтобы внести надписи на диаграмме, необходимо на панели инструментов "Форматирование" зафиксировать пиктограмму Text Tool (символ буквы "А"). Щелкните мышкой на изображении человечка, курсор установится на поле с надписью Actor. Введите в это поле наименование контрагента.

Введите наименование компании "МЕД" в нарисованный прямоугольник, щелкнув мышкой по прямоугольнику. Обратите внимание на то, что при этом должна быть активна пиктограмма Text Tool (символ буквы "А").

Аналогичным образом внесите надписи к линиям соединения фирмы и контрагентов.

Физическая диаграмма ЗАО "МЕД" представлена на [рисунке 2](#).



**Рис. 2.** Физическая диаграмма ЗАО "МЕД"

## 2.2. Задание 2. Формирование списка бизнес-процессов

На основании описания деятельности компании, изложенного в Задании №1, выделите основные бизнес-процессы и занесите их краткое наименование в таблицу со следующим содержанием:

| Номер бизнес-процесса | Наименование бизнес-процесса |
|-----------------------|------------------------------|
|                       |                              |

Номер бизнес-процесса составьте из букв и цифр так, чтобы по номеру был интуитивно понятен смысл бизнес-процесса.

### *Выполнение задания 2*

Для того чтобы выделить бизнес-процессы, необходимо выделить действия, которые совершает компания. В рассматриваемом случае компания планирует закупки, закупает медикаменты, доставляет медикаменты на склад, приходит медикаменты на склад, продает медикаменты. Пример заполнения таблицы бизнес-процессов:

| Номер бизнес-процесса | Название бизнес-процесса |
|-----------------------|--------------------------|
| 1Пл_Зак               | Планирование закупок     |
| 2-Закупк              | Закупки                  |
| 3-Доставк             | Доставка                 |
| 4-Склад               | Запасы-Склад             |

Примечание. В целях упрощения задачи в дальнейшем объединим описание бизнес-процессов "Закупки" и "Планирование закупок" в один бизнес-процесс под названием "Планирование закупок и размещение заказов" и присвоим ему номер 1Пл\_Зак.

## ***Бизнес-процесс "Планирование поставщикам"***

## ***закупок и размещение заказов***

### **Общее описание бизнес-процесса**

Предприятие планирует закупки медикаментов. Планирование закупок осуществляется в Департаменте маркетинга, в группе маркетинга и планирования. Планирование закупок осуществляется следующим образом:

- Менеджер группы планирования и маркетинга ежедневно получает от контрагентов данные внешней и внутренней статистики продаж медикаментов в виде отчетов продаж.

- Для планирования закупок медикаментов менеджер группы планирования и маркетинга еженедельно на основании статистики продаж производит расчет потребности в товаре. В результате расчета формируется Таблица потребностей в товаре.

- Определив количество и номенклатуру заказываемых товаров, менеджер отдела закупок приступает к анализу предложений поставщиков. Данный процесс осуществляется ежемесячно или по мере необходимости. Выбираются наиболее выгодные условия поставки. Для этого сравниваются цены поставщиков. Данные сведения берутся из прайс-листа для закупок. При выборе поставщика важно учесть предоставляемую отсрочку платежа. Эта информация берется из контрактов, отмеченных как приоритетные (действующие). В результате формируется список поставщиков, каждой позиции присваивается признак основного и запасных поставщиков в порядке убывания приоритета.

- Менеджер отдела закупок ежемесячно на основании Таблицы потребностей в товаре и списка выбранных поставщиков формирует графики поставок с указанием сроков и периодичности, но без количества поставки.

- Ежемесячно после определения потребности в товаре менеджер группы логистики рассчитывает необходимое количество закупок. Необходимое количество закупок рассчитывается на основании фактических запасов на складе, необходимого минимального и максимального уровня запасов. Нормы минимального и максимального количества запасов устанавливаются в днях. При расчете необходимого количества закупки учитывается также время товара в пути. Таким образом, данный расчет должен обеспечить возможность бесперебойного отпуска товара со склада. По результату расчетов формируется план заявок на месяц.

- Затем в группе логистики ежедневно по плану заявок, графику поставок, прайс-листам поставщиков формируются заказы поставщикам.
- Если предстоит сделать заказ импортному поставщику, то менеджер группы логистики рассчитывает затраты на сертификацию, создается отчет о затратах на сертификацию. Затраты на сертификацию проверяются на соответствие внутрифирменным нормам. Данная операция производится по мере необходимости.
- Если затраты на сертификацию превышают внутрифирменные нормы, то менеджер группы логистики повторяет процесс формирования заказов поставщикам. Формируются новые заказы.
- Ежедневно подготовленный заказ поставщику акцептуется, заказ должен подписать менеджер по логистике и директор Департамента маркетинга и управления товарными запасами.
- Ежедневно менеджер группы логистики направляет заказ в отдел закупок. Менеджер отдела закупок направляет заказ поставщику.

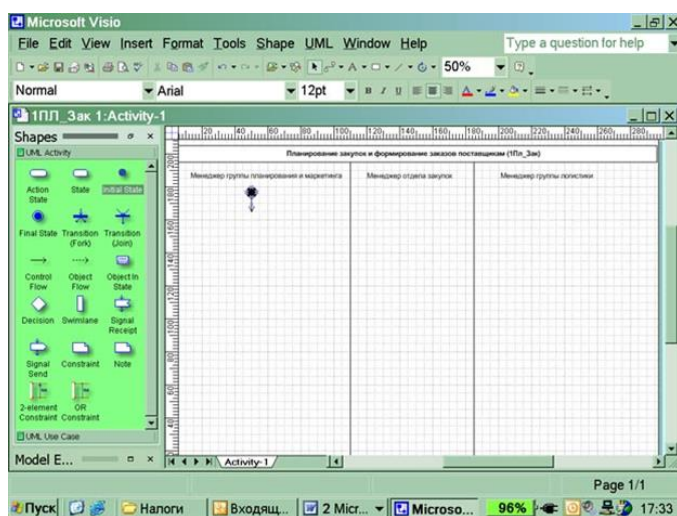
### 2.3. Задание 3. Построение диаграммы действий

На основании общего описания бизнес-процесса "Планирование закупок и размещение заказов поставщикам" составьте диаграмму действий, которая показывает участников процесса, выполняемые каждым участником операции и взаимосвязь между ними. Операции на диаграмме должны следовать в хронологическом порядке, который определен в приведенном описании бизнес-процесса.


#### ***Выполнение задания 3***

- Изучите общее описание бизнес-процесса, выделите его участников. В пунктах №1, 2 приведенного описания участник процесса - "Менеджер группы планирования и маркетинга", в пунктах № 3, 4 - "Менеджер отдела закупок", с 5 по 9 пункт участник бизнес-процесса - "Менеджер группы логистики". Таким образом, в бизнес-процессе "Закупки" три участника - менеджер группы планирования и маркетинга, менеджер отдела закупок, менеджер группы логистики.
- Приступите к формированию диаграммы действий. Для этого необходимо разделить поле на 3 части, каждая часть поля отводится для отображения действий участника процесса.

- Для формирования диаграммы средствами MS Visio необходимо открыть в папке **Software / UML Model Diagram** форму UML Activity.
- Для удобства построения диаграммы на листе расположите его горизонтально (File / Page Setup / Landscape).
- На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением линии Line Tool. Удерживая левую клавишу мыши, разделите лист на три части.
- На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А". Внесите в качестве заголовка полное наименование бизнес-процесса, сокращенное наименование (1Пл\_Зак) и участников бизнес-процесса в соответствии с [рисунком 3](#).

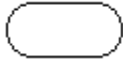


**Рис. 3.** Подготовительная стадия для изображения диаграммы действий


- Проанализируйте общее описание бизнес-процесса и выделите участника процесса, с которого начинается процесс. Очевидно, что это менеджер группы планирования и маркетинга. Действительно, процесс закупок должен начинаться только после того, как определена потребность компании в товаре (медикаментах).
- Обозначьте на диаграмме начало процесса символом  "Initial state" и опустите стрелку вниз ([рис. 3](#)). Работу с графическими формами можно осуществлять только при активированной пиктограмме с изображением стрелки на панели "Форматирование".
- Пользуясь текстовым описанием, выделите действия, выполняемые менеджером группы планирования и маркетинга. Действия (операции), выполняемые менеджером группы планирования и маркетинга: "Получение внутренней статистики продаж", "Получение внешней статистики продаж", "Расчет потребности в товаре".
- Отобразите на диаграмме действия, выполняемые менеджером группы планирования и маркетинга. Обратите внимание, что процессы получения внутренней и




внешней статистики происходят независимо друг от друга. Неважно, в какой последовательности будут получены данные статистики, поэтому действия (операции) по получению внутренней и внешней статистики отобразите на схеме параллельно.

- Для изображения действия на диаграмме используйте фигуру . Впишите внутри фигуры наименование и порядковый номер действия (операции). Пусть параллельные операции имеют номера 1а), 1б). Для ввода текста на панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А".

- Действия соедините на диаграмме стрелками, перенося их мышкой с формы. Стрелки присоединяйте к отмеченным крестиком местам на фигурах.

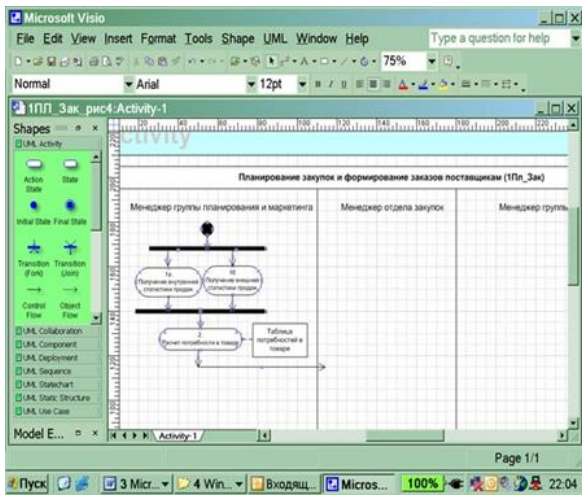
- Для изображения параллельных процессов получения внутренней и внешней статистики примените  (Transition|Fork).

- Расчет потребностей в товаре менеджер выполняет только после того, как получит и внутреннюю, и внешнюю статистику, следовательно, необходимо объединить параллельные процессы получения статистики в один. Для объединения независимых, параллельных процессов используйте  (Transition|Join).

- В результате операции по расчету потребностей в товаре (операция № 2) (п. 2 общего описания) менеджер формирует документ - таблицу потребностей в товаре. Для отображения документа на диаграмме используйте изображение прямоугольника. Нарисуйте прямоугольник мышкой, зафиксировав на панели инструментов "Стандартная" соответствующую пиктограмму Rectangle Tool.

- Операция и получаемый в результате ее выполнения документ на диаграмме соединяются пунктирной линией. Для изображения пунктирной линии зафиксируйте пиктограмму Line Tool на панели инструментов "Стандартная" и выберите пунктирную линию на панели инструментов "Форматирование", используя меню пиктограммы (Line Patter).

- В результате на диаграмме ([рис. 4](#)) получите изображение действий (операций), осуществляемых менеджером группы планирования и маркетинга.



**Рис. 4.** Диаграмма действий менеджера группы планирования и маркетинга

– После того как менеджер группы планирования и маркетинга сформировал таблицу потребностей в товаре, в работу включается менеджер отдела закупок, поэтому направьте стрелку от операции "Расчет потребности в товаре" в поле деятельности менеджера закупок, как показано на [рисунке 4](#).

– Прочитайте общее описание бизнес-процесса и выделите действия (операции), выполняемые менеджером отдела закупок. Определите также действия, которые менеджер отдела закупок выполняет после действий менеджера группы логистики.

– На диаграмме последовательно отобразите следующие действия менеджера отдела закупок:

- Ввод в систему прайс-листов поставщиков (операция № 3)
- Анализ предложений поставщиков (операция № 4)
- Выбор поставщиков (операция № 5)
- Формирование графика поставок без указания количества (операция № 6)
- Осуществите графическое построение диаграммы аналогично описанному в п. 11.
- Соедините действия менеджера отдела закупок стрелками аналогично описанию, приведенному в п. 12.

– Поставьте в соответствие действиям менеджера отдела закупок документы, формируемые в системе. В данном случае это прайс-листы и контракты, список поставщиков с расстановкой приоритетов, график поставок. Выполните работу по рисованию диаграммы в соответствии с описанием в п. 15-16.

– После формирования менеджером отдела закупок графика поставок в работу включается менеджер группы логистики.




– На диаграмме предстоит отобразить следующие действия менеджера группы логистики:


- Расчет необходимого количества закупок (операция № 7);
- Формирование заказов поставщикам (операция № 8);
- Расчет затрат на сертификацию импортных товаров, если медикаменты импортные.\*) (операция № 9);
- Проверка суммы затрат на сертификацию на непревышение внутрифирменной нормы\*);
- Формирование заказов поставщикам при превышении затрат на сертификацию (операция № 10);
- Подпись заказа (операция № 11);
- Направление заказа менеджеру отдела закупок (операция № 12).

Изучая общее описание бизнес-процесса, обратите внимание на то, что менеджер группы логистики дважды производит проверку условий и в зависимости от результата выполняет то или иное действие. В приведенном выше списке операций символом \*) отмечены операции по проверке условий. В этом состоит особенность диаграммирования действий менеджера группы логистики.

– Отобразите действие "Расчет необходимого количества закупок" и опустите стрелку вниз.


– Ввиду того, что формирование заказов поставщикам может происходить неоднократно при превышении затрат на сертификацию, предусмотрите эту ситуацию и используйте графику для объединения параллельных потоков  (Transition|Join).

– Отобразите действие "Формирование заказов поставщикам" после символа объединения потоков.

– Отобразите ромб-символ проверки условия . Проведите из него две стрелки и надпишите их "Импорт", "Россия".


– Стрелку "Россия" направьте к операции № 11 "Подпись заказа".

– По направлению стрелки "Импорт" диаграммируйте последовательно два действия "Расчет затрат на сертификацию импортных товаров", "Проверка суммы затрат на сертификацию на непревышение внутрифирменной нормы".

– За операцией "Проверка суммы затрат на сертификацию на превышение внутрифирменной нормы" вновь отобразите ромб-символ проверки условия . Проведите из него две стрелки и надпишите их "больше x%", "меньше x%". Здесь x% - норма затрат на сертификацию.


– Стрелку с надписью "больше x%" соедините с операцией № 8 "Формирование заказов поставщикам" через символ объединения потоков.

– Стрелку с надписью "меньше x%" направьте к операции № 11 "Подпись заказа".

– Поскольку к операции № 11 "Подпись заказа" направлено два потока действий (п. 29 и п. 33), необходимо воспользоваться обозначением объединения независимых (параллельных) потоков  (Transition|Join). В операцию №11 "Подпись заказа", как и в любую другую, должна входить только одна стрелка. Для выполнения этого правила и используют символ объединения потоков.

– Поставьте в соответствие операции "Подпись заказа" документ - акцептованный заказ поставщику аналогично тому, как написано в п. 15-16.

– В качестве следующей операции отобразите операцию № 12 "Направление заказа менеджеру отдела закупок". На этом действия, выполняемые менеджером группы логистики, завершаются. Вновь работа переключается на менеджера отдела закупок, поэтому направьте стрелку от 12 операции в поле действий менеджера закупок.

– Отобразите на диаграмме переход документа "Заказ поставщику" от менеджера группы логистики к менеджеру отдела закупок. Для этого сначала поставьте в соответствие операции № 12 "Направление заказа менеджеру отдела закупок" документ "Заказ поставщику" так, как это описано в п. 15-16. После этого изображение документа с надписью "Заказ поставщику" путем копирования разместите в поле действий менеджера отдела закупок. Затем направьте пунктирную стрелку  (Object Flow) между двумя отображениями документа "Заказ поставщику" в направлении поля действий менеджера отдела закупок.

– Соедините операцию № 12 "Направление заказа менеджеру отдела закупок" с операцией № 13 "Направление заказа поставщику", выполняемой менеджером отдела закупок. Это последняя операция в соответствии с заданием.

– Укажите на диаграмме конец процесса. Для этого используйте символ ● (Final State). Соедините стрелкой операцию № 13 "Направление заказа поставщику" с символом Final State.

Общий вид диаграммы действий бизнес-процесса "Планирование закупок, формирование заказов поставщикам" представлен на [рис. 5](#).

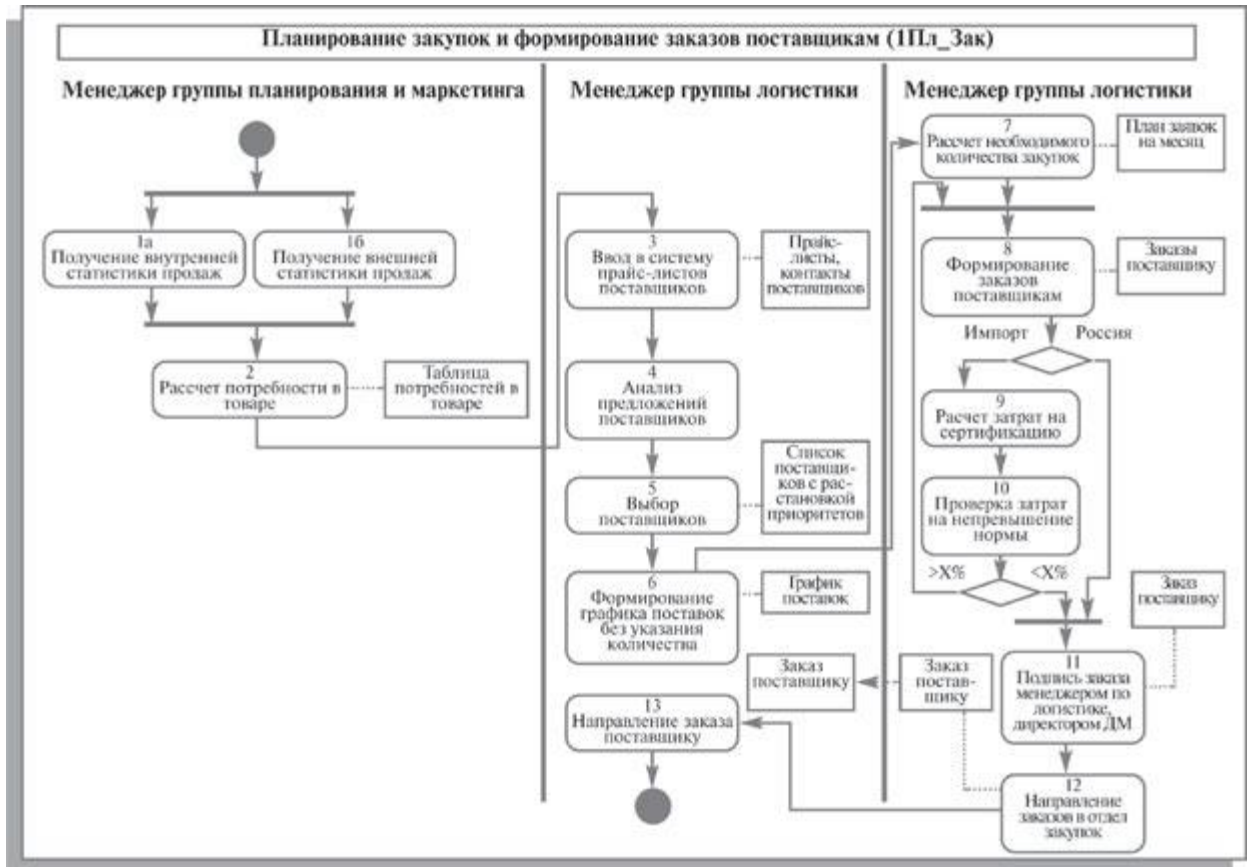


Рис. 5. Диаграмма действий бизнес-процесса "Планирование закупок, формирование заказов поставщикам"

### 3. Практическое занятие 2. Предварительное формирование таблиц

#### 3.1. Задание 4. Формирование таблицы операций

Все операции, участвующие в процессе "Планирование закупок, формирование заказов поставщикам", отразите в Таблице описания операций, имеющей следующий формат:

| Диаграмма и номер на диаграмме | Операция | Исполнитель | Как часто | Входящие документы (документы-основания) | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
|--------------------------------|----------|-------------|-----------|--|--|--|-------------|
| 1                              | 2        | 3           | 4         | 5  | 6  | 7  | 8           |

Примечание. Далее заполненная форма таблицы описания операций будет использоваться для проецирования перечисленных в ней операций на информационную систему.

#### **Выполнение задания 4**

В таблицу последовательно внесите операции бизнес-процесса в соответствии с общим описанием и диаграммой действий.

- В графе 1 проставьте краткое наименование диаграммы - 1Пл\_Зак. Кроме того, в этой графе укажите номер операции, соответствующий изображению на диаграмме действий ([рис. 5](#)).
- В графу 2 путем копирования перенесите из диаграммы действий наименование операции.
- В графе 3 укажите исполнителя операции. В рассматриваемом бизнес-процессе исполнителями операций являются менеджер группы планирования и маркетинга, менеджер отдела закупок, менеджер группы логистики. Графа 3 заполняется на основании диаграммы действий.

– В графе 4 укажите, с какой частотой выполняется каждая операция. Проставьте частоту выполнения операций в соответствии с общим описанием бизнес-процесса. Данная информация фиксируется в ходе обследования компании. Например, это может быть "еженедельно", "ежесуточно", 1 раз в месяц, 200 раз в день и т. п. Если операция выполняется с неопределенной периодичностью, то в графе указывают "по мере необходимости". При проектировании или выборе системы данные из графы "Как часто" определяют требования к быстродействию системы, к параметрам сетевого варианта системы.

– В графу 5 занесите наименования документов, на основании которых осуществляется выполнение операции (входящие документы).

– В графе 6 укажите наименования документов, которые создаются в результате выполнения операции (исходящие документы). В отдельных случаях входящие и исходящие документы могут совпадать. Например, для операции "Направление заказа поставщику" входящим и исходящим документом будет заказ поставщику.

– Если на основании операции формируется бухгалтерская проводка, то она указывается в графе 7. В рассматриваемом примере нет операций, по которым бы формировались проводки.

– Графа 8 предназначена для произвольной дополнительной информации.

На следующих страницах представлена таблица описания операций бизнес-процесса "Планирование закупок и размещение заказов поставщикам".

| Операции бизнес-процесса "Планирование закупок и размещение заказов поставщикам" |                         |              |            |  |  |  |             |
|--|-------------------------|--------------|------------|--|--|--|-------------|
| Диаграмма и номер операции на диаграмме  | Операция                | Исполнитель  | Как часто  | Входящие документы (документы-основания) | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
| 1  | 2                       | 3            | 4          | 5  | 6  | 7  | 8           |
| 1Пл_Зак  | 1. Получение внутренней | Менеджер гр. | Ежесуточно | Отчет-таблица                            | Нет  | Нет  |             |

|               |  |   |                                       |   |  |     |  |
|---------------|--|---|---------------------------------------|---|--|-----|--|
| 1а            | статистики продаж  | планирован ия и маркетинга              |                                       | собственных продаж  |  |     |  |
| 1Пл_Зак<br>16 | 2. Получение внешней статистики продаж                     | Менеджер гр. планирован ия и маркетинга | Ежесут очно                           | Отчет-таблица продаж внешних источников                                     | Нет  | Нет |  |
| 1Пл_Зак<br>2  | 3. Расчет потребностей в товаре                            | Менеджер гр. планирован ия и маркетинга | Ежене дельно                          | Отчет-таблица собственных продаж<br>Отчет-таблица продаж внешних источников | Таблица потребнос тей в товаре                   | Нет |  |
| 1Пл_Зак<br>3  | 4. Ввод в систему прайс-листов поставщиков                 | Менеджер отдела закупок                 | Ежеме сячно                           | Прайс-листы поставщиков   | Прайс-листы поставщик ов                         | Нет |  |
| 1Пл_Зак<br>4  | 5. Анализ предложений поставщиков и действующих контрактов | Менеджер отдела закупок                 | Ежеме сячно и по мере необхо димост и | Прайс-листы поставщиков<br>Контракты действующие                            | Список поставщик ов                              | Нет |  |
| 1Пл_Зак<br>5  | 6. Выбор поставщиков                                       | Менеджер отдела закупок                 | Ежеме сячно и по мере необхо димост и | Список поставщиков  | Список поставщик ов с расстанов кой приоритет ов | Нет |  |
| 1Пл_Зак<br>6  | 7. Формирование графика поставок без указания              | Менеджер отдела закупок                 | Ежеме сячно и по мере необхо          | Список поставщиков с расстановкой приоритетов                               | График поставок                                  | Нет |  |

|               |  |                                 |   |   |   |     |  |
|---------------|--|---------------------------------|---|---|---|-----|--|
|               | количества   |                                 | димост  | Таблица<br>и<br>потребностей в<br>товаре                                      |   |     |  |
| 1Пл_Зак<br>7  | 8. Расчет<br>необходимого<br>количества<br>закупок с<br>учетом остатка<br>на складе и<br>страхового<br>запаса              | Менеджер<br>группы<br>логистики | Ежеме<br>сячно<br>и по<br>мере<br>необхо<br>димост<br>и | Таблица<br>потребностей в<br>товаре,<br>график<br>поставок                    | План<br>заявок на<br>месяц                    | Нет |  |
| 1Пл_Зак<br>8  | 9. Формирование<br>заказов<br>поставщикам с<br>учетом<br>складских<br>остатков,<br>товара в пути<br>и резервного<br>запаса | Менеджер<br>группы<br>логистики | Ежедн<br>евно<br>по<br>плану<br>заявок                  | План заявок<br>на месяц,<br>график<br>поставок,<br>прайс-листы<br>поставщиков | Заказы<br>поставщик<br>у                      | Нет |  |
| 1Пл_Зак<br>9  | 10. Расчет<br>затрат на<br>сертификацию  | Менеджер<br>группы<br>логистики | По<br>мере<br>необхо<br>димост<br>и                     | Заказы<br>поставщику  | Отчет о<br>затратах<br>на<br>сертифика<br>цию | Нет |  |
| 1Пл_Зак<br>10 | 11. Проверка<br>затрат на<br>непревышение<br>нормы   | Менеджер<br>группы<br>логистики | По<br>мере<br>необхо<br>димост<br>и                     | Отчет о<br>затратах на<br>сертифика<br>цию                                    | Отчет о<br>затратах<br>на<br>сертифика<br>цию | Нет |  |
| 1Пл_Зак<br>11 | 12. Подпись<br>заказа<br>менеджером<br>по логистике,<br>директором<br>ДМ   | Менеджер<br>группы<br>логистики | Ежедн<br>евно   | Заказы<br>поставщику  | Заказы<br>поставщик<br>у<br>акцептова<br>нные | Нет |  |

|               |   |                                 |               |   |   |     |  |
|---------------|---|---------------------------------|---------------|---|---|-----|--|
| 1Пл_Зак<br>12 | 13.<br>Направление<br>заказа в отдел<br>закупок | Менеджер<br>группы<br>логистики | Ежедн<br>евно | Заказы<br>поставщику<br>акцептованны<br>е | Заказы<br>поставщик<br>у<br>акцептова<br>нные | Нет |  |
| 1Пл_Зак<br>13 | 14.<br>Направление<br>заказа<br>поставщику      | Менеджер<br>отдела<br>закупок   | Ежедн<br>евно | Заказы<br>поставщику<br>акцептованны<br>е | Заказы<br>поставщик<br>у<br>акцептова<br>нные | Нет |  |

### 3.2. Задание 5. Формирование таблицы описания документов

Все документы, участвующие в бизнес-процессе, отразите в Таблице описания документов, имеющей следующий формат:

| Диаграмм<br>а и номер<br>на<br>диаграмм<br>е | Составляемы<br>й документ<br>(исходящий<br>документ) | Операци<br>я | Кто<br>составляет<br>(исполнитель<br>) | Как<br>част<br>о | Документы<br>-основания<br>(входящие<br>документы<br>) | Реестр, в<br>котором<br>регистрируетс<br>я документ | Комментари<br>й |
|--|--|--------------|--|------------------|--|---|-----------------|
| 1  | 2  | 3            | 4                                      | 5                | 6  | 7   | 8               |

Примечание. После того, как документы будут описаны, приступают к их разработке в ИС. Формы документов в учебном пособии не представлены, в практической же деятельности создается альбом форм, который является приложением к таблице описания документов.

#### **Выполнение задания 5**

Таблица описания документов получается путем переформирования (перестановки столбцов и объединении строк) таблицы описания операций. Особенности таблицы описания документов заключаются в следующем. В Графе 2 не должно быть повторяющихся наименований документов. Если один и тот же документ является исходящим на различных операциях, то он один раз указывается в графе 2 "Составляемый документ", а в графе 3 ему в соответствие ставятся несколько операций. Также по наименованию документа следует



объединить записи и в других графах. В графе 7 указывается наименование реестра, в котором регистрируется создаваемый документ. Наименование реестру присваивается, как правило, по наименованию документа. Например, если документ "Заказ", то "Реестр заказов"; документ "прайс-лист", тогда "реестр прайс-листов" и т.д. На следующих страницах приведена таблица описания документов бизнес-процесса "Планирование закупок и размещение заказов поставщикам".

Бизнес-процесс "Запасы-склад (оприходование товара)"

### Общее описание бизнес-процесса

ЗАО "МЕД" располагает 10 складами, из которых один, Центральны й, расположен в Москве, а другие в филиалах. Количество хранимой номенклатуры медикаментов - от 1000 до 2000.

| Документы бизнес-процесса "Планирование закупок и размещение заказов поставщикам" |  |   |  |                                    |  |   |             |
|---|--|---|--|------------------------------------|--|---|-------------|
| Диаграмма и номер операции на диаграмме   | Составляемый документ (Исходящий документ) | Операция  | Исполнитель                              | Как часто                          | Документы-основания (Входящие документы)             | Реестр, в котором регистрируется документ | Комментарий |
| 1   | 2  | 3   | 4  | 5                                  | 6  | 7   | 8           |
| 1Пл_Зак   | 1. Таблица потребности в товаре            | Расчет потребностей в товаре                          | Менеджер в гр. планирования и маркетинга | Еженедельно                        | Отчет-таблица собственных продаж                     | Реестр статистических отчетов             |             |
| 2   |  |   |  |                                    |  |   |             |
| 1Пл_Зак   | 2. Список поставщиков                      | Анализ предложений поставщиков действующих контрактов | Менеджер отдела закупок                  | Ежемесячно и по мере необходимости | Прайс-листы поставщиков<br><br>Контракты действующие | Реестр прайс-листов                       |             |
| 3   |  |   |  |                                    |  |   |             |

|                  |  |   |                           |                                    |  |                |  |
|------------------|--|---|---------------------------|------------------------------------|--|----------------|--|
| 1Пл_Зак<br>4     | 3. Список поставщиков с расстановкой приоритетов | Выбор поставщиков   | Менеджер отдела закупок   | Ежемесячно и по мере необходимости | Список поставщиков   | Нет            |  |
| 1Пл_Зак<br>5     | 4. График поставок                               | Формирование графика поставок без указания количества   | Менеджер отдела закупок   | Ежемесячно и по мере необходимости | Список поставщиков с расстановкой приоритетов<br><br>Таблица потребностей в товаре | Нет            |  |
| 1Пл_Зак<br>6     | 5. План заявок на месяц                          | Расчет необходимого количества закупок с учетом остатка на складе и страхового запаса           | Менеджер группы логистики | Ежемесячно и по мере необходимости | Таблица потребностей в товаре, прайс-листы поставщиков, график поставок            | Нет            |  |
| 1Пл_Зак<br>7     | 6. Заказы поставщику                             | Формирование заказов поставщикам с учетом складских остатков, товара в пути и резервного запаса | Менеджер группы логистики | Ежедневно по плану заявок          | План заявок на месяц, график поставок, прайс-листы поставщиков                     | Реестр заказов |  |
| 1Пл_Зак<br>9, 10 | 7. Отчет о затратах на сертификацию              | Расчет затрат на сертификацию<br><br>Проверка   | Менеджер группы логистики | По мере необходимости              | Заказы поставщику  | Нет            |  |

|                           |  |  |                                 |               |   |                   |  |
|---------------------------|--|--|---------------------------------|---------------|---|-------------------|--|
|                           |  | затрат на<br>непревышение<br>нормы   |                                 |               |   |                   |  |
| 1Пл_Зак<br><br>11, 12, 13 | 8. Заказы<br>поставщику<br>акцептованн<br>ые | Подпись заказа<br>менеджером по<br>логистике,<br>директором<br>ДМ<br><br>Направление<br>заказа в отдел<br>закупок<br><br>Направление<br>заказа<br>поставщику | Менеджер<br>группы<br>логистики | Ежедне<br>вно | Заказы<br>поставщик<br>у<br><br>Заказы<br>поставщик<br>у<br>акцептован<br>ные | Реестр<br>заказов |  |

Склад фактически работает не с номенклатурой, а с сериями. Одной позиции номенклатуры может соответствовать несколько серий медикаментов.

Склад разбит на несколько зон хранения. Зоны хранения соответствуют правилам хранения тех или иных медикаментов.

Используются вложенные единицы измерения - упаковка (минимальная единица), блок (несколько упаковок), заводская коробка (несколько блоков).

На складе хранится товар зарезервированный (недоступный для продажи).

Учет ТМЦ ведется в двух валютах - в рублях, валюте прихода.

Процесс оприходования медикаментов на склад выглядит следующим образом:

- Менеджер приемного отдела принимает товар по товарной накладной поставщика, проверяя номенклатуру, количество, посерийное соответствие, срок годности.
- При полном соответствии фактически поступившего товара товару, указанному в товарно-транспортной накладной и заказе поставщику, менеджер приемного отдела передает

документы менеджеру отдела закупок. В противном случае осуществляется процесс выявления виновных лиц и предъявление претензий.

- Менеджер отдела закупок проверяет соответствие поставки заказу по номенклатуре, количествам и ценам и на основании товарной накладной поставщика формирует приходную накладную, отражая в базе данных количество и учетную цену поступившего товара. При формировании приходной накладной создается проводка Д41-К60. Далее в работу включаются менеджеры отделов сертификации и маркетинга.

- Менеджер отдела сертификации по товарно-транспортной накладной проверяет наличие серий в справочнике. При необходимости справочник серий пополняется.

- Менеджером отдела сертификации осуществляется процесс сертификации. Процесс сертификации в данном случае рассматривается и как процесс приходования сертификатов-документов на медикаменты, и как процедура сертификации с целью получения документов-сертификатов.

- Менеджер учетного отдела при приходовании ТМЦ по товарно-транспортной накладной разбивает каждую номенклатурную позицию по сериям с указанием срока годности.

- Параллельно с работой менеджера по сертификации, после отражения в базе данных количества товара менеджером отдела закупок, менеджер отдела маркетинга, используя товарно-транспортную накладную, определяет базовую цену продажи и указывает ее в карточке товара.

- Размещение товара по местам хранения осуществляется менеджером склада в соответствии с Планом расстановки продукции по местам хранения. Место хранения заносится в карточку товара.

### 3.3. Задание 6. Построение диаграммы действий

На основании общего описания бизнес-процесса "Запасы-склад (приходование)" составьте диаграмму действий, которая показывает участников процесса, выполняемые каждым участником операции и взаимосвязь между ними. Операции на диаграмме должны следовать в хронологическом порядке, который определен в приведенном описании бизнес-процесса.

#### ***Выполнение задания 6***

– Изучите общее описание бизнес-процесса, выделите его участников. В пунктах № 1, 2 приведенного описания участник процесса - "Менеджер отдела приемки", в пункте № 3 - участник "Менеджер отдела закупок", в пункте № 4, 7 - участник "Менеджер отдела сертификации", в пункте № 5 - участник бизнес-процесса "Менеджер учетного отдела", в пункте № 6 - "Менеджер отдела маркетинга", в пункте № 8 - "Менеджер склада".

Таким образом, в бизнес-процессе "Запасы-Склад" шесть участников процесса - менеджеры отделов приемки, закупок, сертификации, учетного отдела, отдела маркетинга и склада.

– Приступите к формированию диаграммы действий. Для этого необходимо разделить поле на 6 частей, каждая из которых отводится для отображения действий участника процесса.


– Для формирования диаграммы средствами MS Visio необходимо открыть в папке **Software / UML Model Diagram** форму UML Activity.


– Для удобства размещения диаграммы на листе расположите его горизонтально ( File / Page Setup / Landscape).

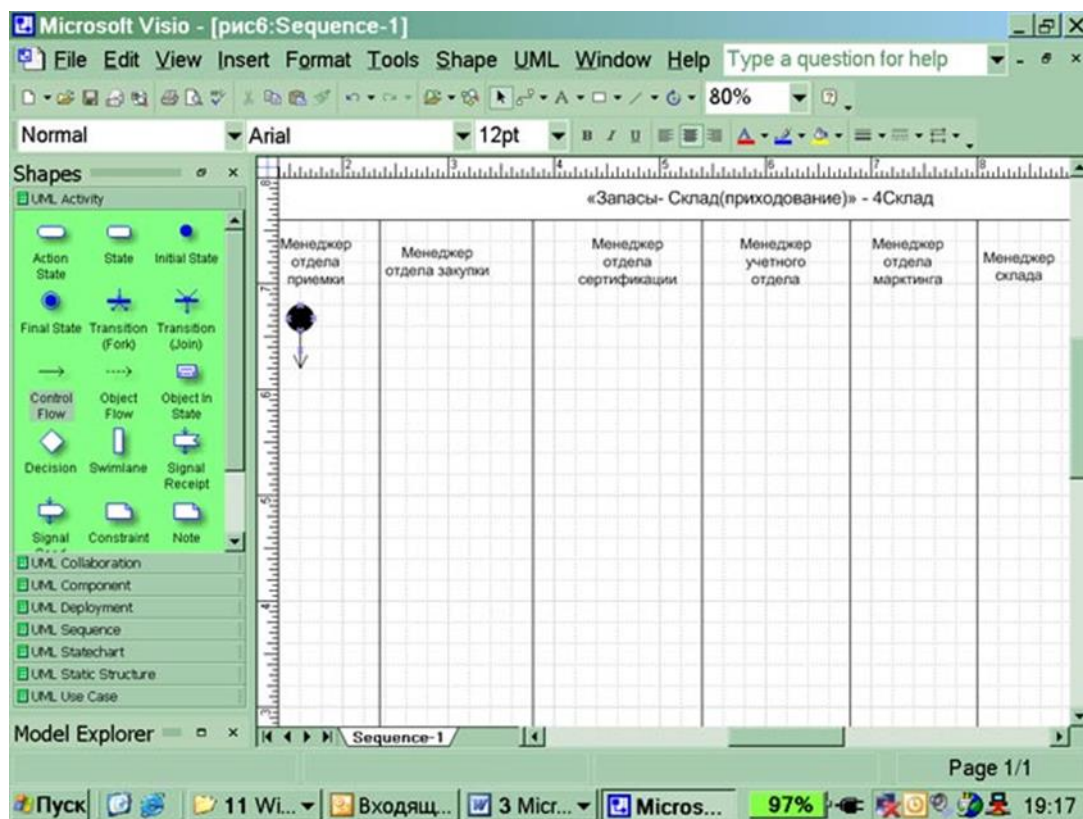
– На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением линии Line Tool. Удерживая левую клавишу мышки, разделите лист на шесть частей.

– На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А". Внесите в качестве заголовка полное наименование бизнес-процесса "Запасы-склад (приходование)", сокращенное наименование (4Склад) и участников бизнес-процесса в соответствии с [рисунком 6](#).



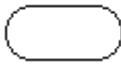
– Проанализируйте общее описание бизнес-процесса и выделите участника процесса, с которого начинается процесс. Очевидно, что это менеджер отдела приемки.

– Обозначьте на диаграмме начало процесса символом  "Initial state" в графе, отведенной для изображения действий менеджера отдела приемки ([рис. 6](#)). Не забывайте, что работу с графическими формами можно осуществлять только при активированной пиктограмме с изображением стрелки на панели "Форматирование".

– Направьте стрелку вниз от изображения начала процесса. Для этого перенесите с формы UML Activity изображение стрелки  (Control Flow) ([Рис. 6](#)).



**Рис. 6.** Подготовительная стадия для изображения диаграммы действий

- Пользуясь текстовым описанием, выделите действия, выполняемые менеджером отдела приемки. Действия (операции), выполняемые менеджером отдела приемки - "проверка товара по количеству, серийному соответствию, сроку годности".
- Отобразите на диаграмме действие, выполняемое менеджером отдела приемки. Для изображения действия на диаграмме используйте фигуру . Впишите внутрь фигуры наименование и порядковый номер (№ 1) операции. Для ввода текста на панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А".
- Отобразите ромб-символ проверки условия . Проведите из него две стрелки и надпишите их "Несоответствие документам", "Полное соответствие документов".
- Стрелку "Несоответствие документам" соедините с подпроцессом выявления виновных лиц и предъявления претензий. Для отображения подпроцесса используйте тот же символ, что и для отображения действия .
- При несоответствии документов заданным требованиям приходования товара на склад не происходит, процесс заканчивается. Проведите стрелку из подпроцесса выявления

виновных лиц и предъявления претензий и  
процесса ●(Final State)

соедините ее с символом завершения

– Стрелку с надписью "Полное соответствие документов" соедините с операцией № 2 "Отражение в базе данных количества товара", выполняемой менеджером отдела закупки.

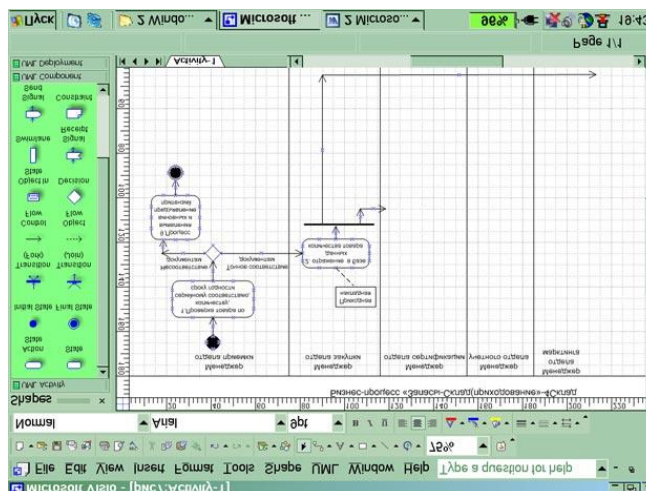
– Отражение в базе данных количества товара осуществляется путем создания в базе приходной накладной. Для отображения документа на диаграмме используйте изображение прямоугольника. Нарисуйте прямоугольник мышкой, зафиксировав на панели инструментов "Стандартная" соответствующую пиктограмму Rectangle Tool.

– Операция № 2 "Отражение в базе данных количества товара" и получаемый в результате ее выполнения документ "Приходная накладная" на диаграмме соединяются пунктирной линией. Для изображения пунктирной линии зафиксируйте пиктограмму Line Tool на панели инструментов "Стандартная" и выберите пунктирную линию на панели инструментов "Форматирование", используя меню пиктограммы (Line Patter).

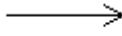
– В соответствии с общим описанием бизнес-процесса (пункты № 3, 7), после выполнения операции № 2 "Отражение в базе данных количества товара", выполняемой менеджером отдела закупки, происходит параллельная работа менеджера отдела сертификации и менеджера отдела маркетинга.

– Для изображения параллельных процессов примените  (Transition|Fork).


– Сравните полученное изображение с [рис. 7](#).



**Рис. 7.** Диаграмма действий менеджера отдела приемки и менеджера отдела закупки

– Отобразите последовательно действия (операции) менеджера отдела сертификации и соедините их стрелками  (Control Flow). Менеджер выполняет операции № 3

"Поиск серии в справочнике", № 4 "Добавление серии в справочник", а также № 5 "Процесс сертификации".

– После выполнения процесса сертификации к работе подключается менеджер учетного отдела. Он выполняет операцию № 6 "Разбиение каждой позиции номенклатуры по сериям". Отобразите эту операцию в зоне действий менеджера учетного отдела, используя, как всегда, символ .



– Изобразите на диаграмме операцию № 7 "Процесс размещения серии товаров", выполняемую менеджером склада.

– Соедините стрелкой операцию № 6, выполняемую менеджером учетного отдела, и № 7, выполняемую менеджером склада.

– Разместите на диаграмме операцию № 8 "Определение и ввод базовой цены", выполняемую менеджером отдела маркетинга.

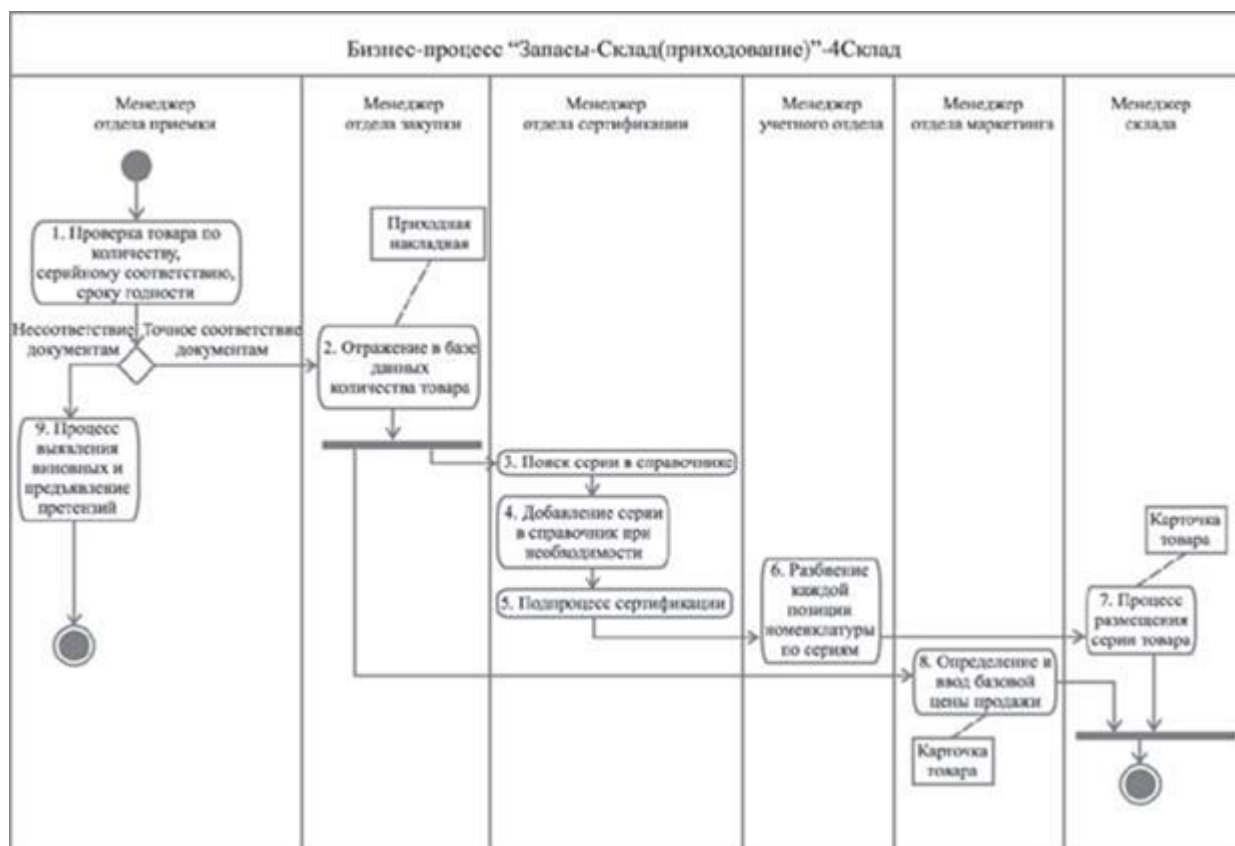
– Соедините операцию № 2 с операцией № 8, используя ранее подготовленную стрелку ([рис. 7](#)).

– Поставьте в соответствие операции №8 документ "Карточка товара" аналогично описанию, приведенному в пунктах 16, 17.

– Операции, выполняемые участниками рассматриваемого процесса, завершены. Отобразите завершение процесса в поле действия менеджера склада. Для этого прежде всего необходимо объединить параллельные операции. Для объединения независимых параллельных процессов используйте  ( Transition|Join). После объединения процессов укажите на диаграмме конец процесса. Для этого используйте символ  (Final State).

– Общий вид диаграммы действий бизнес-процесса "Запасы - Склад (приходование)" представлен на [рис. 8](#).





**Рис. 8.** Диаграмма действий бизнес-процесса "Запасы - Склад (приходование)"

## Лабораторная работа 2. Формирование таблицы операций

### 3.4. Задание 7. Формирование таблицы операций

Все операции, участвующие в процессе "Запасы - Склад (приходование)", отразите в Таблице описания операций, имеющей следующий формат:

| Диаг<br>рамма и<br>номер<br>на диаграмме | Оп<br>ерация | Испол<br>нитель | ак<br>часто | Входя<br>щие<br>документы<br>(документы-<br>основания) | Исходящ<br>ий документ<br>(составляемый<br>документ) | Пров<br>одка<br>(дебет,<br>кредит,<br>сумма,<br>аналитика) | Комме<br>нтарий |
|--|--------------|-----------------|-------------|--|--|--|-----------------|
| 1  | 2            | 3               |             | 5  | 6  | 7  | 8               |

Примечание. Далее заполненная форма таблицы описания операций будет использоваться для проектирования перечисленных в ней операций в информационной системе.

### Выполнение задания 7

В таблицу последовательно внесите операции бизнес-процесса в соответствии с общим описанием и диаграммой действий.

- В графе 1 проставьте краткое наименование диаграммы - 4Склад. Кроме того, в этой графе укажите номер операции, соответствующий изображению на диаграмме действий ([рис. 8](#)).

- В графу 2 путем копирования перенесите из диаграммы действий наименование операции.

- В графе 3 укажите исполнителя операции. В рассматриваемом бизнес-процессе исполнителями операций являются менеджер отдела приемки, отдела закупки, сертификации, учетного отдела, отдела маркетинга, склада. Графа 3 заполняется на основании диаграммы действий.

- В графе 4 укажите, с какой частотой выполняется каждая операция. Проставьте частоту выполнения операций в соответствии с общим описанием бизнес-процесса. В данном примере все операции за исключением одной выполняются ежедневно. Только операция № 4 выполняется нерегулярно - по мере необходимости. Данная информация фиксируется в ходе обследования компании. При проектировании или выборе системы данные из графы "Как часто" определяют требования к быстродействию системы, к параметрам сетевого варианта системы.

- В графу 5 занесите наименования документов, на основании которых осуществляется выполнение операции (входящие документы). В данном примере основанием выполнения почти всех операций является товарно-транспортная накладная. Кроме того, основанием операции № 2 является также Заявка поставщику, при выполнении операции № 3 используется справочник серий, а основанием процесса сертификации (операция № 5) служит Сертификат.

- В графе 6 укажите наименования документов, которые создаются в результате выполнения операции (исходящие документы). В данном бизнес-процессе только два исходящих (формируемых) документа. Это приходная накладная и карточка товара.

- Если на основании операции формируется бухгалтерская проводка, то она указывается в графе 7. В рассматриваемом примере проводка формируется по операции с номером 2.

- Графа 8 предназначена для произвольной дополнительной информации.

Ниже представлена таблица описания операций бизнес-процесса.

| Операции бизнес-процесса "Запасы - склад (приходование)" |  |                              |                       |   |  |  |             |
|--|--|------------------------------|-----------------------|---|--|--|-------------|
| Диаграмма и номер на диаграмме                           | Операция   | Исполнитель                  | Как часто             | Входящие документы (документы - основания)            | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
| 1  | 2  | 3                            | 4                     | 5   | 6  | 7  | 8           |
| 4Склад<br>1  | 1. Проверка товара по количеству, серийному соответствию, сроку годности | Менеджер приемного отдела    | Ежедневно             | Товарная накладная поставщика                         |  |  |             |
| 4Склад<br>2  | 2. Отражение в базе данных количества и цены приходования товара         | Менеджер отдела закупок      | Ежедневно             | Товарная накладная поставщика<br><br>Заказ поставщику | Приходная накладная                        | Д41-К60                                    |             |
| 4Склад<br>3  | 3. Поиск серии в справочнике   | Менеджер отдела сертификации | Ежедневно             | Справочник серий<br><br>Товарная накладная поставщика |  |  |             |
| 4Склад<br>4  | 4. Добавление серии в справочник   | Менеджер отдела сертификации | По мере необходимости | Товарная накладная поставщика                         | Запись в справочнике серий                 | Нет  |             |

|             |   |  |               |                                     |                    |  |  |
|-------------|---|--|---------------|-------------------------------------|--------------------|--|--|
| 4Склад<br>5 | 5. Подпроцесс<br>сертификации                               | Менеджер<br>отдела<br>сертификаци<br>и | Ежедн<br>евно | Сертификат                          |                    |  |  |
| 4Склад<br>6 | 6. Разбиение<br>каждой позиции<br>номенклатуры<br>по сериям | Менеджер<br>учетного<br>отдела         | Ежедн<br>евно | Товарная<br>накладная<br>поставщика |                    |  |  |
| 4Склад<br>8 | 7. Определение<br>и ввод базовой<br>цены продажи            | Менеджер<br>отдела<br>маркетинга       | Ежедн<br>евно | Товарная<br>накладная<br>поставщика | Карточка<br>товара |  |  |
| 4Склад<br>7 | 8. Процесс<br>размещения<br>серии товара                    | Менеджер<br>склада                     | Ежедн<br>евно |                                     | Карточка<br>товара |  |  |

### 3.5. Задание 8. Формирование таблицы описания документов

Все документы, участвующие в бизнес-процессе, отразите в Таблице описания документов, имеющей следующий формат:

| Диаграмм<br>а и номер<br>на<br>диаграмм<br>е | Составляемы<br>й документ<br>(исходящий<br>документ) | Операци<br>я | Кто<br>составляет<br>(исполнитель<br>) | Как<br>част<br>о | Документы<br>-основания<br>(входящие<br>документы<br>) | Реестр, в<br>котором<br>регистрируетс<br>я документ | Комментари<br>й |
|--|--|--------------|--|------------------|--|---|-----------------|
| 1  | 2  | 3            | 4                                      | 5                | 6  | 7   | 8               |

Примечание. После того как будут описаны документы, приступают к их разработке в ИС. Формы документов в учебном пособии не представлены, в практической же деятельности создается альбом форм, который является приложением к таблице описания документов.

#### **Выполнение задания 8**

Таблица описания документов создается путем переформирования (перестановки столбцов и объединении строк) таблицы описания операций. Особенности таблицы описания документов заключаются в следующем. В Графе 2 не должно быть повторяющихся наименований документов. Если один и тот же документ является исходящим на различных

операциях, то он один раз указывается в графе 2 "Составляемый документ", а в графе 3 ему в соответствие ставятся несколько операций. Также по наименованию документа следует объединить записи и в других графах.

В графе 7 указывается наименование реестра, в котором регистрируется создаваемый документ. Наименование реестру присваивается, как правило, по наименованию документа. В данном бизнес-процессе приходная накладная регистрируется в реестре приходных накладных, карточка товара регистрируется в реестре товаров.

Ниже приведена таблица описания документов бизнес-процесса "Запасы - Склад (приходование)".

| Документы бизнес-процесса "Запасы - Склад (приходование)" |  |  |                              |                       |   |   |             |
|---|--|--|------------------------------|-----------------------|---|---|-------------|
| Диаграмма и номер на диаграмме                            | Составляемый документ (исходящий документ) | Операция                                     | Кто составляет (исполнитель) | Как часто             | Документы-основания (входящие документы)              | Реестр, в котором регистрируется документ | Комментарий |
| 1   | 2  | 3  | 4                            | 5                     | 6   | 7   | 8           |
| 4Склад<br>2   | 1. Приходная накладная                     | 1. Отражение в базе данных количества товара | Менеджер отдела закупок      | Ежедневно             | Товарная накладная поставщика<br><br>Заказ поставщику | Реестр приходных накладных                |             |
| 4Склад<br>4   | 2. Запись в справочнике серий              | 2. Добавление серии в справочник             | Менеджер отдела сертификации | По мере необходимости | Товарная накладная поставщика                         | Нет                                       |             |
| 4Склад<br>7, 8  | 3. Карточка товара                         | 3. Определение и ввод базовой                | Менеджер отдела маркетинга   | Ежедневно             | Товарная накладная поставщика                         | Реестр товара                             |             |

|  |  |                                    |                 |  |  |  |  |
|--|--|------------------------------------|-----------------|--|--|--|--|
|  |  | цены продажи                       | га              |  |  |  |  |
|  |  | 4. Процесс размещения серии товара | Менеджер склада |  |  |  |  |

### ***Бизнес-процесс "Продажи"***

#### **Общее описание бизнес-процесса**

Бизнес-процесс выглядит следующим образом:

- Менеджер отдела продаж ежедневно получает от клиента Заказ на конкретную номенклатурную единицу медикаментов. В Заказе номенклатурных единиц клиент указывает желаемую отсрочку платежа.
- При получении Заказа менеджер отдела продаж по справочнику лицензий проверяет наличие у клиента действующей лицензии на право реализации медикаментов. При отсутствии лицензии продажа медикаментов клиенту не производится. Наличие лицензии проверяется по мере необходимости.
- Менеджер отдела продаж ежедневно проверяет наличие необходимого количества заказанных медикаментов на складе.
- Если медикаментов недостаточно для выполнения заказа, то менеджер отдела продаж размещает Заказ в реестре "неудовлетворенный спрос". Затем менеджер ежедневно проверяет возможность выполнения Заказа, размещенного в реестре "неудовлетворенный спрос".
- При наличии у клиента необходимой лицензии и достаточном количестве товара на складе в отделе продаж на основании Заказа и договора формируется Заявка на номенклатурные единицы. Заявки формируются ежедневно.
- Ежедневно на основании Заявки менеджер отдела продаж осуществляет резервирование товара.
- Менеджер отдела продаж ежедневно контролирует кредитный лимит и дебиторскую задолженность потенциальных покупателей.
- Если кредитный лимит и дебиторская задолженность не превышают допустимых значений, то Заявка передается на склад в Учетно-операционный отдел.

- При превышении кредитного лимита или наличии просроченной дебиторской задолженности свыше допустимого количества дней менеджер отдела продаж заявку в Учетно-операционный отдел не передает, процесс продаж приостанавливается, осуществляются переговоры с клиентом.

- Менеджер учетно-операционного отдела, получив Заявку, ежедневно производит подборку номенклатурных единиц.

- Менеджер учетно-операционного отдела ежедневно формирует упаковочные листы для вложения их в каждый ящик.

- Менеджером учетно-операционного отдела ежедневно формируются для клиента следующие документы: счет, расходная накладная, счет-фактура.

- При фактической отгрузке товара со склада осуществляется его списание. Списание медикаментов осуществляется по расходной накладной и сопровождается формированием проводки Д62-К41.

### 3.6. Задание 9. Построение диаграммы действий

На основании общего описания бизнес-процесса "Продажи" составьте диаграмму действий, которая показывает участников процесса, выполняемые каждым участником операции и взаимосвязь между ними. Операции на диаграмме должны следовать в хронологическом порядке, который определен в приведенном описании бизнес-процесса.

#### **Выполнение задания 9**

- Изучите общее описание бизнес-процесса, выделите его участников. Участниками бизнес-процесса "Продажи" являются менеджер отдела продаж и менеджер учетно-операционного отдела.

- Приступите к формированию диаграммы действий. Для этого необходимо разделить поле на 2 части, каждая из которых отводится для отображения действий участника процесса.

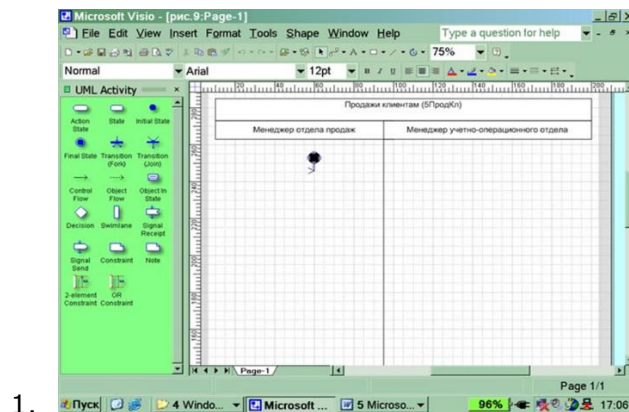
- Для формирования диаграммы средствами MS Visio необходимо открыть в папке **Software / UML Model Diagram** форму UML Activity.

- Для удобства построения диаграммы на листе расположите его вертикально (File/Page Setup/Portrait).

– На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением линии Line Tool. Удерживая левую клавишу мыши, разделите лист на две части.

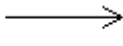
– На панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А". Внесите в качестве заголовка полное наименование бизнес-процесса "Продажи клиентам", сокращенное наименование (5ПродКл) и участников бизнес-процесса в соответствии с [рисунком 9](#). Проанализируйте общее описание бизнес-процесса и выделите участника процесса, с которого начинается процесс. Очевидно, что это менеджер отдела продаж.

– Обозначьте на диаграмме начало процесса символом ● "Initial state" в графе, отведенной для изображения действий менеджера отдела продаж ([рис. 9](#)). Не забывайте, что работу с графическими формами можно осуществлять только при активированной пиктограмме с изображением стрелки на панели "Форматирование".



1.

– Рис. 9. Подготовительная стадия для изображения диаграммы действий

– Направьте стрелку вниз от изображения начала процесса. Для этого перенесите с формы UML Activity изображение стрелки  (Control Flow) (Рис. 9).

– Пользуясь текстовым описанием, выделите действия, выполняемые менеджером отдела продаж. Действия (операции), выполняемые менеджером отдела продаж:

– Операция № 1 "Получение от клиента заказа с указанием номенклатурных единиц товара по количеству, серийному соответствию, сроку годности"

– Операция №2 "Проверка наличия у клиента лицензии на заказанные медикаменты";

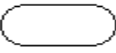
– Операция № 3 "Проверка наличия товарных запасов на складе";

– Операция № 4 "Размещение заказа в реестре неудовлетворенный спрос";

– Операция № 5 "Процесс формирования заявки на основании заказа и договора";




- Операция № 6 "Резервирование товара";
- Операция № 7 "Контроль кредитного лимита и дебиторской задолженности";
- Операция № 8 "Отклонение заявки".
- Отобразите на диаграмме первые две операции, выполняемые менеджером отдела


продаж. Для изображения действия на диаграмме используйте фигуру 


- Впишите внутри двух фигур  наименования и порядковые номера операций.

Для ввода текста на панели инструментов "Стандартная" зафиксируйте пиктограмму с изображением буквы "А".


- Соедините операции в порядке их следования стрелками  (Control Flow).

– Отобразите ромб-символ проверки условия . Проведите из него две стрелки и надпишите их "Лицензия есть", "Лицензии нет". Стрелку с надписью "Лицензии нет" предстоит позже соединить с символом конца бизнес-процесса.

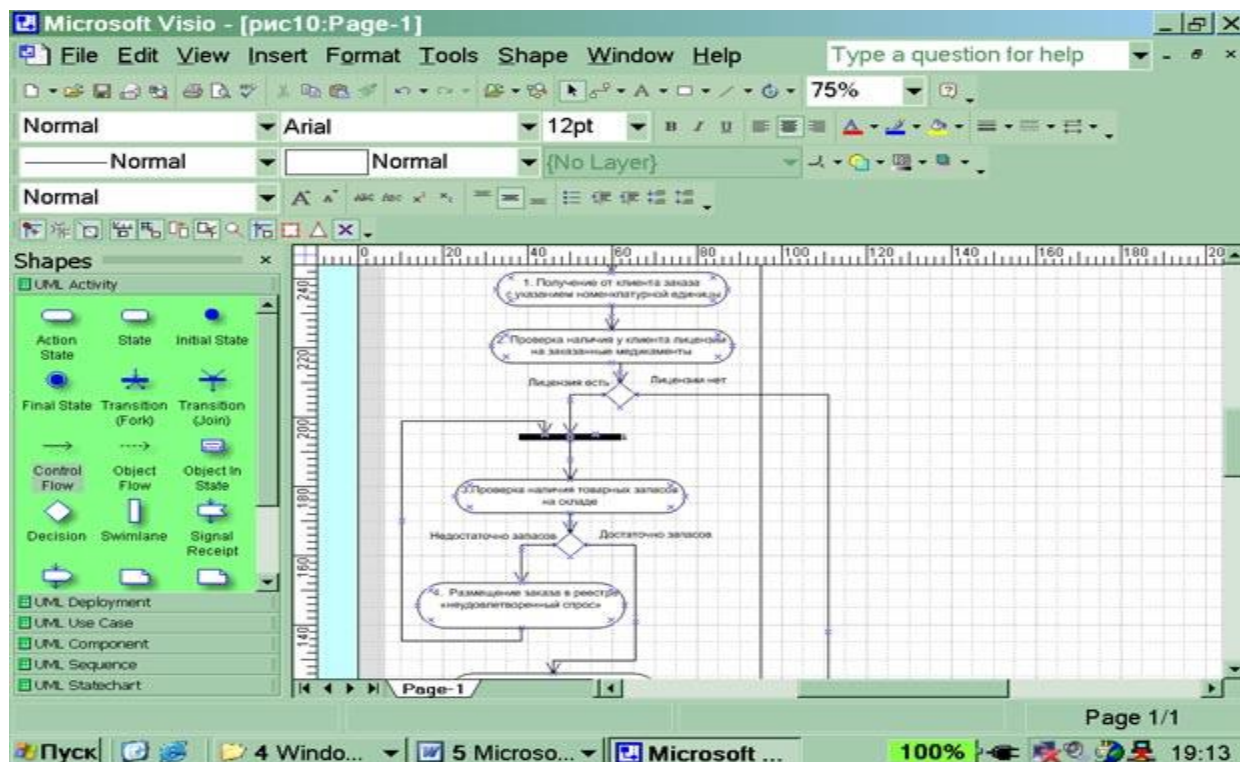
– Стрелку "Лицензия есть" соедините с операцией № 3 "Проверка наличия товарных запасов на складе", для изображения которой примените символ 

– Отобразите ромб-символ проверки условия . Проведите из него две стрелки и надпишите их "Достаточно запасов", "Недостаточно запасов".

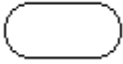
– Стрелку с надписью "Недостаточно запасов" соедините с операцией № 4 "Размещение заказа в реестре "неудовлетворенный спрос".

– От операции № 4 направьте стрелку к операции № 3, для того чтобы отразить циклический процесс контроля выполнения отложенных заявок. Обратите внимание на то, что к операции № 3 уже направлена стрелка с надписью "Лицензия есть". Поскольку по правилам построения диаграмм в операцию может входить только одна стрелка, воспользуйтесь символом объединения независимых потоков  (Transition|Join).

– Сверьте созданный фрагмент диаграммы действий бизнес-процесса "Продажи" с рисунком 10.




**Рис. 10.** Фрагмент диаграммы действий бизнес-процесса "Продажи"

– Стрелку с надписью "Достаточно запасов" направьте к операции № 5 "Процесс формирования заявки на основании заказа и договора", для изображения которой используйте, как всегда, символ .

– Поставьте в соответствие операции № 5 документ "Заявка". Для отображения документа на диаграмме используйте изображение прямоугольника. Нарисуйте прямоугольник мышкой, зафиксировав на панели инструментов "Стандартная" соответствующую пиктограмму Rectangle Tool.

– Операция № 5 и получаемый в результате ее выполнения документ "Заявка" соединяются на диаграмме пунктирной линией. Для изображения пунктирной линии зафиксируйте пиктограмму Line Tool на панели инструментов "Стандартная" и выберите пунктирную линию на панели инструментов "Форматирование", используя меню пиктограммы (Line Patter). Затем мышкой нарисуйте соединение операции с документом.

– За операцией № 5 последовательно отобразите операции № 6 "Резервирование товара" и № 7 "Контроль кредитного лимита и дебиторской задолженности".

– Операцию № 7 соедините с ромбом  - символом проверки условия. Проведите из него две стрелки и надпишите их "Есть превышения", "Нет превышений".

– Стрелку с надписью "Есть превышения" соедините с операцией № 8 "Отклонение заявки". В последующем операцию № 8 предстоит соединить с символом завершения бизнес-процесса. Заметьте, что на завершение бизнес-процесса уже направлена стрелка с надписью "Лицензии нет".



– На этом действия менеджера отдела продаж завершаются, и к работе подключается менеджер учетно-операционного отдела. В поле действий менеджера учетно-операционного отдела последовательно отобразите выполняемые им операции: № 9 "Подбор номенклатурных единиц", № 10 "Формирование упаковочных листов", № 11 "Формирование счета, расходной накладной, счета-фактуры", № 12 "Отгрузка и списание медикаментов".

– Соедините стрелками операции № 9, 10, 11, 12.

– К операции № 9 подведите стрелку с надписью "Нет превышений".

– Поставьте в соответствие операции № 10 документ "Упаковочный лист", операции № 11 - три документа: "Счет", "Расходная накладная", "Счет-фактура". Для отображения документа на диаграмме используйте изображение прямоугольника. Нарисуйте прямоугольник мышкой, зафиксировав на панели инструментов "Стандартная" соответствующую пиктограмму Rectangle Tool.

– Операции № 10 и № 11 и получаемые в результате их выполнения документы соедините пунктирной линией. Для изображения пунктирной линии зафиксируйте пиктограмму Line Tool на панели инструментов "Стандартная" и выберите пунктирную линию на панели инструментов "Форматирование", используя меню пиктограммы (Line Patter). Затем мышкой нарисуйте соединение операции с документом.

– Операция № 12 была последней в бизнес-процессе "Продажи", поэтому остается отобразить на диаграмме действий символ завершения процесса. На символ завершения бизнес-процесса направлено три стрелки, отображающие независимые процессы. Примените обозначение для объединения независимых процессов  (Transition|Join). После объединения трех процессов укажите на диаграмме конец процесса. Для этого используйте символ  (Final State).

Общий вид диаграммы действий бизнес-процесса "Продажи" представлен [рис. 11](#).

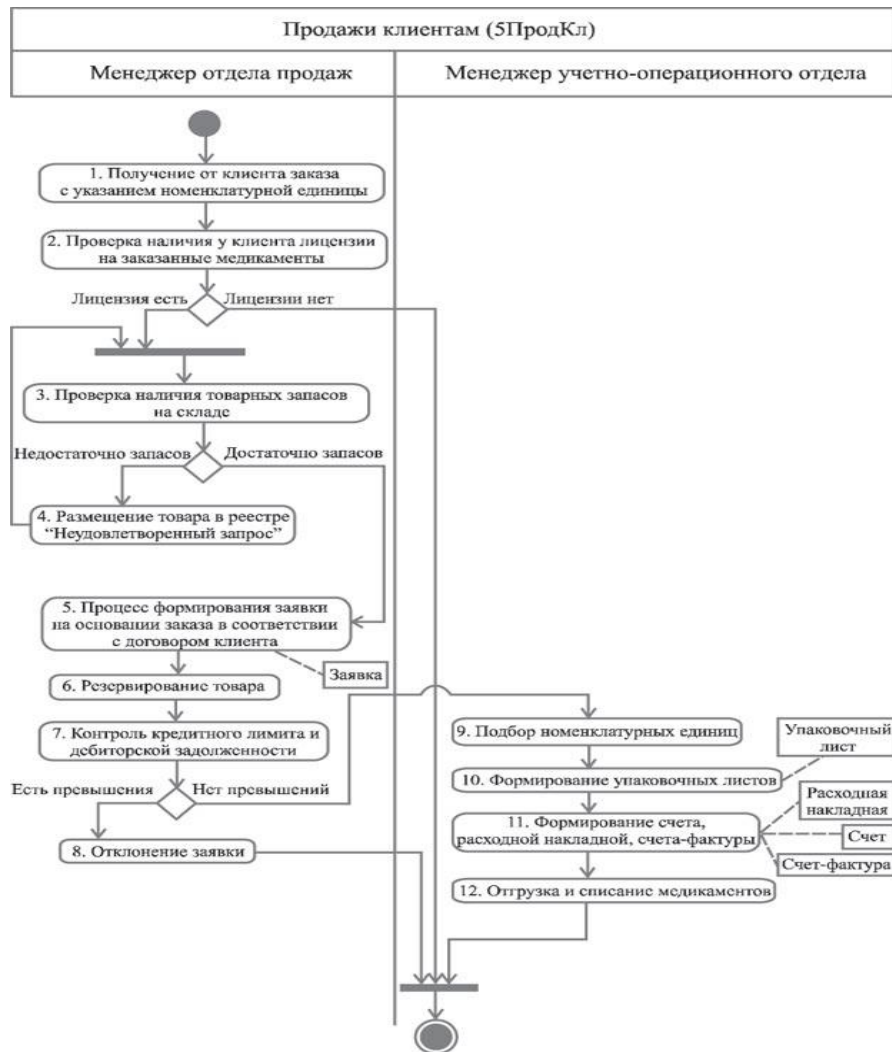


Рис. 11. Диаграмма действий бизнес-процесса "Продажи"

## 4. Лабораторная работа 2. Формирование таблицы операций

### 4.1. Задание 10. Формирование таблицы операций

Все операции, участвующие в процессе "Продажи", отразите в Таблице описания операций, имеющей следующий формат:

| Диаграмма и номер на диаграмме | Операция | Исполнитель | Как часто | Входящие документы (документы-основания) | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
|--------------------------------|----------|-------------|-----------|--|--|--|-------------|
| 1                              | 2        | 3           | 4         | 5  | 6  | 7  | 8           |

Примечание. Далее заполненная форма таблицы описания операций будет использоваться для проектирования перечисленных в ней операций в информационной системе.

#### **Выполнение задания 10**

В таблицу последовательно внесите операции бизнес-процесса в соответствии с общим описанием и диаграммой действий.

- В графе 1 проставьте краткое наименование диаграммы - 5Прод\_Кл. Кроме того, в этой графе в каждой строке укажите номер операции, соответствующий изображению на диаграмме действий ([рис. 11](#)).

- В графу 2 путем копирования перенесите из диаграммы действий наименование операций.

- В графе 3 укажите исполнителя операции. В бизнес-процессе "Продажи" исполнителями операций являются менеджеры отдела продаж и учетно-операционного отдела. Графа 3 заполняется на основании диаграммы действий.

- В графе 4 укажите, с какой частотой выполняется каждая операция. Проставьте частоту выполнения операций в соответствии с общим описанием бизнес-процесса. В данном примере все операции, кроме операций № 2 и № 4, выполняются ежедневно. Не регулярно, по мере необходимости выполняются операции № 2, 4. Данная информация фиксируется в

ходе обследования компании. При проектировании или выборе системы данные из графы "Как часто" определяют требования к быстродействию системы и к параметрам сетевого варианта системы.

– В графу 5 занесите наименования документов, на основании которых осуществляется выполнение операции (входящие документы). В данном примере основанием выполнения операций № 1-5 является Заказ номенклатурных единиц. Кроме того, основанием операции № 2 является также Справочник лицензий клиента, для операции № 3 - Картотека склада. Документ "Заявка на номенклатурные единицы" является основанием операций № 6, 7, 9, 11, 12. Для выполнения операции № 7 используются также Приказ по компании о кредитном лимите и отчет о дебиторской задолженности. Операция № 12 требует также наличия Расходной накладной.

– В графе 6 в соответствии с общим описанием укажите наименования документов, которые создаются в результате выполнения операций (исходящие документы). В данном бизнес-процессе исходящими (формируемыми) документами являются Заявка номенклатурных единиц, Реестр "неудовлетворенный спрос", Упаковочный лист, Счет, Расходная накладная, Счет-фактура.

– Если на основании операции формируется бухгалтерская проводка, то она указывается в графе 7. В рассматриваемом примере проводка формируется по операции с номером 12.

– Графа 8 предназначена для произвольной дополнительной информации.

Ниже представлена таблица описания операций бизнес-процесса "Продажи".

| Описание операций бизнес-процесса "Продажи" |   |              |           |  |  |  |             |
|---|---|--------------|-----------|--|--|--|-------------|
| Диаграмма и номер на диаграмме              | Операция  | Исполнитель  | Как часто | Входящие документы (документы-основания) | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
| 1   | 2   | 3            | 4         | 5  | 6  | 7  | 8           |
| 1ПродКл<br>1                                | 1. Получение от клиента заказа с указанной номенклатурной | Отдел продаж | Ежедневно | Заказ номенклатурных единиц              |  |  |             |

|              |  |              |                       |  |                                  |  |   |
|--------------|--|--------------|-----------------------|--|----------------------------------|--|---|
| 1ПродКл<br>2 | единицей<br>2. Проверка наличия у клиента лицензии на заказанные медикаменты                           | Отдел продаж | По мере необходимости | Заказ номенклатурных единиц<br>Справочник лицензий клиента |                                  |  |   |
| 1ПродКл<br>3 | 3. Проверка наличия товарных запасов на складе   | Отдел продаж | Ежедневно             | Картотека склада<br>Заказ номенклатурных единиц            |                                  |  |   |
| 1ПродКл<br>4 | 4. Размещение заказа в реестре "неудовлетворенный спрос" при частичном или полном не выполнении заказа | Отдел продаж | По мере необходимости | Заказ номенклатурных единиц                                | Реестр "Неудовлетворенный спрос" |  | В реестре указывается заказ с неудовлетворенным спросом |
| 1ПродКл<br>5 | 5. Процесс формирования заявки на основании заказа в соответствии с договором клиента                  | Отдел продаж | Ежедневно             | Заказ номенклатурных единиц                                | Заявка на номенклатурные единицы |  | В заявку копируются из договора скидки, наценки         |
| 1ПродКл<br>6 | 6. Резервирование товара   | Отдел продаж | Ежедневно             | Заявка на номенклатурные единицы                           |                                  |  |   |
| 1ПродКл<br>7 | 7. Проверка кредитного лимита и дебиторской задолженности  | Отдел продаж | Ежедневно             | Заявка на номенклатурные единицы                           |                                  |  |   |

|               |  |                           |           |                                    |   |         |  |
|---------------|--|---------------------------|-----------|------------------------------------|---|---------|--|
|               |  |                           |           | Приказ о кредитном лимите          |   |         |  |
|               |  |                           |           | Отчет по дебиторской задолженности |   |         |  |
| 1ПродКл<br>9  | 8. Подбор заказанных номенклатурных единиц                 | Учетно-операционный отдел | Ежедневно | Картотека складского учета         |   |         |  |
|               |  |                           |           | Заявка на номенклатурные единицы   |   |         |  |
| 1ПродКл<br>10 | 10. Формирование упаковочных листов                        | Учетно-операционный отдел | Ежедневно | Заявка на номенклатурные единицы   | Упаковочный лист                            |         |  |
| 1ПродКл<br>11 | 11. Формирование счета, расходной накладной, счета-фактуры | Учетно-операционный отдел | Ежедневно | Заявка на номенклатурные единицы   | Счет<br>Счет-фактура<br>Расходная накладная |         |  |
| 1ПродКл<br>12 | 11. Отгрузка и списание медикаментов                       | Учетно-операционный отдел | Ежедневно | Заявка на номенклатурные единицы   | Расходная накладная                         | Д62-К41 |  |

#### 4.2. Задание 11. Формирование таблицы описания документов

Все документы, участвующие в бизнес-процессе, отразите в Таблице описания документов, имеющей следующий формат:

| Диаграмма | Составляемые | Операция | Кто | Как | Документы- | Реестр, | в Коммен |
|-----------|--------------|----------|-----|-----|------------|---------|----------|
|-----------|--------------|----------|-----|-----|------------|---------|----------|



| и номер на<br>диаграмме | й документ<br>(исходящий<br>документ) |   | составляет<br>(исполнитель) | часто | основания<br>(входящие<br>документы) | котором<br>регистрируе<br>тся<br>документ | тарий |
|-------------------------|---------------------------------------|---|-----------------------------|-------|--------------------------------------|---|-------|
| 1                       | 2                                     | 3 | 4                           | 5     | 6                                    | 7   | 8     |

Примечание. После того как будут описаны документы, приступают к их разработке в ИС. Формы документов в учебном пособии не представлены, в практической же деятельности создается альбом форм, который является приложением к таблице описания документов.

### **Выполнение задания 11**

Таблица описания документов создается путем переформирования (перестановки столбцов и объединения строк) таблицы описания операций. Особенности таблицы описания документов заключаются в следующем. В Графе 2 не должно быть повторяющихся наименований документов. Если один и тот же документ является исходящим на различных операциях, то он один раз указывается в графе 2 "Составляемый документ", а в графе 3 ему в соответствие ставятся несколько операций. Так же по наименованию документа следует объединить записи и в других графах.

В графе 7 указывается наименование реестра, в котором регистрируется создаваемый документ. Наименование реестру присваивается, как правило, по наименованию документа. В данном бизнес-процессе расходная накладная регистрируется в реестре расходных накладных, счет - в реестре счетов, счет-фактура - в реестре счетов-фактур, заявка - в реестре заявок.

На следующих страницах приведена таблица описания документов бизнес-процесса "Продажи".

**Описание документов бизнес-процесса "Продажи".**

| Диаграмма<br>и номер<br>операции<br>на<br>диаграмме | Составляе<br>мый<br>документ<br>(исходящи<br>й<br>документ) | Входящие<br>документы<br>(документы-<br>основания) | Операция  | Исполнитель  | Как<br>часто                 | Реестр, в<br>котором<br>регистри<br>руется<br>документ | Комм<br>ентар<br>ий |
|---|---|--|---|--------------|------------------------------|--|---------------------|
| 5ПродКл<br>4  | 1. Реестр<br>"Неудовлетворе<br>нный спрос"                  | Заказ<br>номенклатурны<br>х единиц                 | Размещение<br>заказа в реестре<br>"неудовлетворен<br>ный спрос" при | Отдел продаж | По мере<br>необход<br>имости | Реестр<br>приходны<br>х<br>накладных                   |                     |

|               |  |   |   |                                  |               |                                  |  |
|---------------|--|---|---|----------------------------------|---------------|----------------------------------|--|
|               |  |   | частичном или<br>полном<br>невыполнении<br>заказа   |                                  |               |                                  |  |
| 5ПродКл<br>5  | 2. Заявка на<br>номенклатурны<br>е единицы | Заказ<br>номенклатурны<br>х единиц      | Процесс<br>формирования<br>заявки на<br>основании<br>заказа в<br>соответствии с<br>договором<br>клиента | Отдел продаж                     | Ежеднев<br>но | Реестр<br>заявок                 |  |
| 5ПродКл<br>10 | 3. Упаковочный<br>лист                     | Заявка на<br>номенклатурны<br>е единицы | Формирование<br>упаковочных<br>листов   | Учетно-<br>операционный<br>отдел | Ежеднев<br>но | Нет                              |  |
| 5ПродКл<br>11 | 4. Счет                                    | Заявка на<br>номенклатурны<br>е единицы | Формирование<br>счета   | Учетно-<br>операционный<br>отдел | Ежеднев<br>но | Реестр<br>счетов                 |  |
| 5ПродКл<br>11 | 5. Счет-<br>фактура                        | Заявка на<br>номенклатурны<br>е единицы | Формирование<br>счета-фактуры   | Учетно-<br>операционный<br>отдел | Ежеднев<br>но | Реестр<br>счетов-<br>фактур      |  |
| 5ПродКл<br>11 | 6. Расходная<br>номенклатура               | Заявка на<br>номенклатурны<br>е единицы | Формирование<br>расходной<br>номенклатуры   | Учетно-<br>операционный<br>отдел | Ежеднев<br>но | Реестр<br>расходных<br>накладных |  |

### ***Бизнес-процесс "Взаиморасчеты с клиентами"***

#### **Общее описание бизнес-процесса**

Бизнес-процесс выглядит следующим образом:

1. Менеджер отдела продаж до 10 раз в день отгружает товары клиентам в соответствии с договорами и Приказом по кредитной линии. Одновременно с отгрузкой товара менеджер отдела продаж выставляет счет клиенту. Счет регистрируется в реестре счетов.
2. По факту произведенной отгрузки менеджер отдела продаж делает запись в журнале отгрузок и оплат, тем самым фиксируя задолженность клиента.
3. Бухгалтер компании ежедневно получает и обрабатывает выписки с расчетных счетов банков. Бухгалтер на основании банковской выписки определяет оплаченные счета и делает отметку об оплате счета в реестре счетов.
4. Менеджер отдела продаж ежедневно контролирует поступление платежей от клиентов, проверяя допустимый срок оплаты счета.
5. Если платежи по счету на расчетный счет компании не поступили и срок оплаты счета истек, то менеджер отдела продаж блокирует отгрузку товара клиенту. Если клиент оплатил счет, то менеджер вносит сведения об оплате в Журнал отгрузок и оплат.
6. Бухгалтер в конце каждого месяца выводит сальдо взаиморасчетов с клиентами.

#### 4.3. Задание 12. Построение диаграммы действий

На основании общего описания бизнес-процесса "Взаиморасчеты с клиентами" составьте диаграмму действий, которая показывает участников процесса, выполняемые каждым участником операции и взаимосвязь между ними. Операции на диаграмме должны следовать в хронологическом порядке, который определен в приведенном описании бизнес-процесса.

#### 4.4. Задание 13. Формирование таблицы операций

Все операции, участвующие в процессе "Взаиморасчеты с клиентами" отразите в Таблице описания операций, имеющей следующий формат:

| Диаграмма и номер на диаграмме | Составляемый документ (исходящий документ) | Операция | Кто составляет (исполнитель) | Как часто | Документы-основания (входящие документы) | Реестр, в котором регистрируется документ | Комментарий |
|--------------------------------|--|----------|------------------------------|-----------|--|---|-------------|
| 1                              | 2  | 3        | 4                            | 5         | 6  | 7   | 8           |

#### 4.5. Задание 14. Формирование таблицы описания документов

Все документы, участвующие в бизнес-процессе, отразите в Таблице описания документов, имеющей следующий формат:

| Диаграмма и номер на диаграмме | Операция | Исполнитель | Как часто | Входящие документы (документы-основания) | Исходящий документ (составляемый документ) | Проводка (дебет, кредит, сумма, аналитика) | Комментарий |
|--------------------------------|----------|-------------|-----------|--|--|--|-------------|
| 1                              | 2        | 3           | 4         | 5  | 6  | 7  | 8           |

*Бизнес-процесс "Взаиморасчеты с поставщиками"*

#### Общее описание бизнес-процесса

Бизнес-процесс выглядит следующим образом:

– Менеджер отдела закупок ежедневно получает от поставщика медикаментов счет на оплату, регистрирует его в реестре счетов поставщиков и передает счет поставщика бухгалтеру.

– Бухгалтер на основании счета поставщика ежедневно формирует платежное поручение на оплату и передает платежное поручение в банк.

– Бухгалтер на основании выписки с расчетного счета банка делает отметку об оплате счета в реестре счетов поставщика.

– Менеджер отдела закупок при поступлении товара и (или) при оплате делает запись в Журнале поступлений и оплат.

– Бухгалтер в конце каждого месяца выводит сальдо взаиморасчетов с клиентами.

## 5. Практическое занятие 3. Построение диаграммы действий.

### 5.1. Задание 15. Построение диаграммы действий

На основании общего описания бизнес-процесса "Взаиморасчеты с поставщиками" составьте диаграмму действий, которая показывает участников процесса, выполняемые каждым участником операции и взаимосвязь между ними. Операции на диаграмме должны следовать в хронологическом порядке, который определен в приведенном описании бизнес-процесса.

### 5.2. Задание 16. Формирование таблицы операций

Все операции, участвующие в процессе "Продажи", отразите в Таблице описания операций, имеющей следующий формат:

| Диаграмма<br>и номер на<br>диаграмме | Составляемый<br>документ<br>(исходящий<br>документ) | Операция | Кто<br>составляет<br>(исполнитель) | Как<br>часто | Документы-<br>основания<br>(входящие<br>документы) | Реестр, в<br>котором<br>регистрируется<br>документ | Коммент<br>арий |
|--------------------------------------|---|----------|------------------------------------|--------------|--|--|-----------------|
| 1                                    | 2   | 3        | 4                                  | 5            | 6  | 7  | 8               |

### 5.3. Задание 17. Формирование таблицы описания документов

#### *Спецификации настроек типовой ИС*

Функциональность модулей ERP-системы MBS Ахарта.

#### **Логистика**

Прогноз закупок, продаж, запасов.

Описание номенклатуры с использованием трех аналитик.

Специальные цены, скидки для номенклатуры со специальной группой аналитики.

Описание хранения с использованием склада, палет и размещения.

Отслеживание номенклатур по серийному номеру и номеру партии

ABC-анализ по заданным пользователем критериям ABC-анализа по реализации, себестоимости и марже.

Управление карантинном. Просмотр номенклатуры на карантинном складе на любом этапе контроля качества.

Поддержка штрих-кодов.

### **Сводное планирование**

Расчет потребности в материалах и мощностях.

Прогнозы закупок и продаж. Возможность обзора долгосрочных потребностей по закупке, производству и ресурсам.

Возможность расчета краткосрочных потребностей на основе существующих заказов и/или прогнозного планирования.

Получение сводного плана по заказам.

Система может предложить внести следующие изменения к существующим и спланированным заказам: Увеличение количества заказа, Уменьшение количества заказа, отложить выполнение заказа или закупки

### **Управление продажами**

Определение планов продаж для менеджеров по продажам и групп менеджеров (отделов продаж)

Управление процессом продаж

Прогнозирование продаж

Отслеживание статуса продаж, включая рассматриваемые предложения

Отслеживание действий и прогресса в работе отдельных сотрудников отдела продаж

Графическое представление данных по продажам

Создание отчетов по предложениям, деятельности менеджеров по продажам и отдельным сотрудникам

### **Торговля**

Учет и размещение номенклатуры на складе

Создание закупок напрямую из заказа

Пересчет единиц измерения по закупке в единице учета на складе

Обработка недопоставок

Автоматическая замена товаров, которых нет в наличии на складе, на другие альтернативные товары

### **Управление складом**

Регистрация и размещение товара, возможность хранения товара в соответствии со структурой склада

Идентификация физического размещения: склад, ячейка и палеты

Идентификация истории происхождения номенклатуры: серийный номер и номер партии

Характеристики товара: конфигурация, цвет и размер

Возможность маркировки как отдельной номенклатуры, так и группы номенклатур с целью дальнейшего отслеживания

Ведение журналов приемки

Возможность перехода из заказов на отгрузку в ячейки комплектации через журналы отгрузки

### **Учет договоров**

Ведение юридической информации о договорах с клиентами и поставщиками, условиях оплаты, контактах и ответственных

Привязка накладных и оплат к конкретному договору (указание договора в строках журналов ГК, заказах, закупках, накладных и оплатах с последующим переносом в проводку по клиенту/поставщику)

Включение атрибутов договоров в предложения по оплате

Автоматическое/периодическое сопоставление проводок по контрагентам и договорам

Форма ручного сопоставления в рамках договоров

Сальдо расчетов в рамках отдельного договора

Номер договора в проводках по курсовой разнице

Переход от моделей предметной области к функциональной модели системы

## 6. Лабораторная работа 3. Проектирование реализации операций бизнес-процесса в информационной системе (ИС)

Бизнес-процесс "Планирование закупок и размещение заказов поставщикам"

### 6.1.Задание 18. Проектирование реализации операций бизнес-процесса в информационной системе (ИС)

Все операции, участвующие в процессе, отразите в Таблице проектирования операций, имеющей следующий формат:

| Номер операции на диаграмме | Операция | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
|-----------------------------|----------|------------------------|---------------------|-----------------------------------|
| 1                           | 2        | 3                      | 4                   | 5                                 |

#### *Выполнение задания 18*

– В графе 1 укажите номер операции и краткое наименование диаграммы действий проектируемого бизнес-процесса. Данные в графу введите в соответствии с таблицей описаний операций.

– В графу 2 перенесите наименования операций из таблицы описания операций.

– В графе 3 перечислите необходимые разработки для реализации операций.

– В графе 4 сформулируйте специфику настройки функционала системы.

– В графе 5 укажите наименование модуля или функции, необходимые для реализации операции бизнес-процесса.

Пример проектирования операций бизнес-процесса "Планирование закупок и размещение заказов поставщикам" в ИС приведен в таблице.

#### Проектирование реализации операций бизнес-процесса в ИС

| Номер операции на диаграмме | Операция | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
|-----------------------------|----------|------------------------|---------------------|-----------------------------------|
| 1                           | 2        | 3                      | 4                   | 5                                 |



|                 |  |   |  |   |
|-----------------|--|---|--|---|
| 1а<br>(1Пл_Зак) | 1. Получение внутренней статистики                               | Разработка узла хранения данных статистики продаж   | Коды клиентов в файле соответствуют кодировке в Системе  | Продажи, клиенты                          |
|                 |  | Разработка механизма импорта статистики   | Единицы измерения номенклатуры соответствуют единицам измерения в Системе<br>Коды номенклатуры статистики соответствуют кодам номенклатуры Системы |   |
| 16<br>(1Пл_Зак) | 2. Получение внешней статистики продаж                           | Разработка узла хранения данных статистики продаж   | Коды клиентов в файле соответствуют кодировке в Системе  | Продажи, клиенты                          |
|                 |  | Разработка механизма импорта статистики   | Единицы измерения номенклатуры соответствуют единицам измерения в Системе  |   |
|                 |  |   | Коды номенклатуры статистики соответствуют кодам номенклатуры Системы  |   |
| 2<br>(1Пл_Зак)  | 3. Расчет потребностей в товаре                                  | Разработка механизма автоматического формирования минимального и максимального запаса препаратов (ассортиментный план на период планирования) эффективности закупок (ABC и XYZ классификации) |  | Сводное планирование, логистика, торговля |
| 3<br>(1Пл_Зак)  | 4. Регистрация прайс-листов поставщиков в системе                | Разработка механизма импорта электронной версии прайс-листа в форму "коммерческого соглашения"  | В системе регистрируется один базовый прайс-лист, на его основе формируются все другие прайс-листы   | Коммерческие соглашения                   |
| 4,<br>(1Пл_Зак) | 5. Анализ прайс-листов поставщиков действующих контрактов. Выбор | Разработка механизма реализации в системе и оценки эффективности закупки на основании полученных прайс-листов, с учетом   |  | Продажи, клиенты                          |

|                 |  |   |  |   |  |
|-----------------|--|---|--|---|--|
|                 | поставщиков приоритетных запасных каждой позиции   | условий и (скидки, по платежа)  | поставки отсрочка  |   |  |
| 6<br>(1Пл_Зак)  | 6. Формирование (регистрация) графика поставок (сроки, переодичность) без указания количеств         | Разработка графика поставок (календаря рабочего времени) для каждого поставщика                 | В качестве графика поставок (график обращений) используется календарь рабочего времени для каждого поставщика  | Календарь рабочего времени                |  |
| 7,<br>(1Пл_Зак) | 8 7. Формирование заказов поставщикам с учетом складских остатков, товара в пути и резервного запаса | Разработка взаимосвязанных таблиц Заказов, Складских остатков, Товара в пути, Резервных заказов | При заполнении в заказе поля "Количество" система в первую очередь "просматривает" количество товаров на складе. При недостаточном количестве товаров на складе система обращается к таблице с данными о резервных запасах. При недостаточном количестве резервных запасов система осуществляет поиск заданной в заказе номенклатуры в таблице Товары в пути | Сводное планирование, логистика, торговля |  |
| 9<br>(1Пл_Зак)  | 8. Расчет затрат на сертификацию   | Разработать механизм расчета затрат на сертификацию при формировании рабочего прайс-листа       | Расчет затрат на сертификацию производится перед формированием прайс-листа поставщика. При формировании рабочего прайс-листа помимо учета скидок, отсрочек платежа, времени движения товара в пути (в денежном выражении), также необходимо учесть стоимость сертификации  | Коммерческие соглашения                   |  |
| 10<br>(1Пл_Зак) | 9. Проверка суммы затрат на сертификацию на превышение нормы   |   | Разработка алгоритма проверки суммы затрат на сертификацию на превышение внутрифирменной нормы   | Сводное планирование, логистика, торговля |  |
| 11<br>(1Пл_Зак) | 10. Подпись заказа   | Разработать процедуру   | Результатом процедуры "Сводное планирование" в форме   | Сводное планирование,                     |  |

|              |  |   |
|--------------|--|---|
|              | менеджером по утверждения строк        | "Спланированные заказы" логистика, торговля   |
|              | логистике, спланированных заказов      | являются строки. После оценки строки в форме спланированные заказы необходимо провести процедуру одобрения (утверждения) строк спланированных заказов |
| 12 (1Пл_Зак) | 11. Направление заказа в отдел закупок | Разработка многопользовательской системы, прав доступа к документам   |
|              |  | Сводное планирование, логистика, торговля   |

### Бизнес-процесс "Запасы-склад (оприходование товара)"

## 6.2.Задание 19. Проектирование реализации операций бизнес-процесса в информационной системе (ИС)

Все операции, участвующие в процессе, отразите в Таблице проектирования операций, имеющей следующий формат:

| Номер операции на диаграмме | Операция | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
|-----------------------------|----------|------------------------|---------------------|-----------------------------------|
| 1                           | 2        | 3                      | 4                   | 5                                 |

### Выполнение задания 19

– В графе 1 укажите номер операции и краткое наименование диаграммы действий проектируемого бизнес-процесса. Данные в графу введите в соответствии с таблицей описаний операций.

– В графу 2 перенесите операции из таблицы описания операций.

– В графе 3 перечислите необходимые разработки для реализации операций.

– В графе 4 сформулируйте специфику настройки функционала системы.

– В графе 5 укажите наименование модуля или функции, необходимые для реализации операции бизнес-процесса.

Пример проектирования операций бизнес-процесса "Планирование закупок и размещение заказов поставщикам" в ИС приведен в таблице.

## Бизнес-процесс "Продажи"

### 6.3.Задание 20. Проектирование реализации операций бизнес-процесса в информационной системе (ИС)

Все операции, участвующие в процессе, отразите в Таблице проектирования операций, имеющей следующий формат:

| Номер операции на диаграмме | Операция | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
|-----------------------------|----------|------------------------|---------------------|-----------------------------------|
| 1                           | 2        | 3                      | 4                   | 5                                 |

#### Выполнение задания 20

– В графе 1 укажите номер операции и краткое наименование диаграммы действий проектируемого бизнес-процесса. Данные в графу введите в соответствии с таблицей описаний операций.

– В графу 2 перенесите операции из таблицы описания операций.

– В графе 3 перечислите необходимые разработки для реализации операций.

– В графе 4 сформулируйте специфику настройки функционала системы.

– В графе 5 укажите наименование модуля или функции, необходимые для реализации операции бизнес-процесса.

Пример проектирования операций бизнес-процесса "Продажи" в ИС приведен в таблице.

| Планирование закупок и размещение заказов поставщиками |   |                        |                     |                                   |
|--|---|------------------------|---------------------|-----------------------------------|
| Номер операции на диаграмме                            | Операция  | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
| 1  | 2   | 3                      | 4                   | 5                                 |
| 4Склад<br>1  | 1. Проверка товара по количеству, серийному соответствию, | Не требуется           |                     | Не требуется                      |

|             |   |   |   |                         |
|-------------|---|---|---|-------------------------|
| 4Склад<br>2 | сроку годности<br>2. Отражение в базе данных количества и учетной цены товара | Разработать электронную и печатную формы приходной накладной и взаимосвязь с финансовым блоком для формирования проводки  | В системе формируются проводки, а также финансовый приход товара на склад                               | Логистика, торговля     |
| 4Склад<br>3 | 3. Поиск серии в справочнике  | Разработать справочник серий и систему поиска в нем   | Перед тем, как создать новый серийный номер, необходимо произвести поиск серийного номера в базе данных | Логистика               |
| 4Склад<br>4 | 4. Добавление серии в справочник  | Разработать функцию в пополнения справочника  |   | Логистика               |
| 4Склад<br>6 | 5. Разбиение каждой позиции номенклатуры по сериям                            | Разработать иерархический номенклатурный справочник   |   | Логистика               |
| 4Склад<br>7 | 6. Процесс размещения серии товара на складе                                  | Разработать разбиение мест хранения по зонам<br>Разработать функцию автоматического подбора зоны хранения<br>Разработать форму карточки товара с возможностью указания в ней места хранения | Настроить аналитику серийный номер обязательную для заполнения при реализации                           | Логистика               |
| 4Склад<br>8 | 7. Определение и ввод базовой цены товара                                     | Разработать базовый прайс-лист  | Базовый прайс-лист формируется на основании данных отдела маркетинга                                    | Коммерческие соглашения |

Бизнес-процесс "Взаиморасчеты с клиентами и поставщиками"

Самостоятельно выполните следующее задание.

#### 6.4.Задание 21. Проектирование реализации операций бизнес-процесса в информационной системе

Все операции, участвующие в процессе, отразите в Таблице проектирования операций, имеющей следующий формат:

| Номер операции на диаграмме | Операция | Необходимые разработки | Специфика настройки | Функциональность (модуль) системы |
|-----------------------------|----------|------------------------|---------------------|-----------------------------------|
| 1                           | 2        | 3                      | 4                   | 5                                 |

Дальнейшая настройка типовой ИС (выбор экранных форм, формирование отчетов и пр.) осуществляется с использованием специфических средств, предусмотренных в каждой системе.

**Пример проектирования операций бизнес-процесса "Продажи"**

| Номер операции и на диаграмме | Операция  | Необходимые разработки  | Специфика настройки   | Функциональность (модуль) системы |
|-------------------------------|---|---|---|-----------------------------------|
| 1                             | 2   | 3   | 4   | 5                                 |
| 5ПродКл-1                     | Получение от клиента заказа с указанной номенклатурной единицей                   | Разработать шаблон импорта заявки клиента.<br>Разработать функционал импорта заявки клиента в заказы          | Шаблон может быть разработан на основе "MS Excel", "MS Word"                              | Логистика, расчеты с клиентами    |
| 5ПродКл-2                     | Проверка наличия у клиента лицензии на заказанные медикаменты                     | Разработать функционал, позволяющий при импорте заявки клиента в систему проверять наличие у клиента лицензии | Каждому клиенту ставится в соответствие лицензия. В лицензии указывается срок ее действия | Логистика                         |
| 5ПродКл-3                     | Проверка наличия товарных запасов на складе для полного или частичного выполнения | Разработать функционал, проверяющий наличие товарных запасов при импорте заявки клиента                       | При импорте заявки система проверяет наличие товара на складе                             | Логистика, сводное планирование   |

|            |  |  |  |                               |
|------------|--|--|--|-------------------------------|
|            | заказа   |  |  |                               |
| 5ПродКл-4  | Размещение заказа в реестре "неудовлетворенный спрос"                              | Разработать функционал, который при импорте заказа клиента и на складе, система отсутствия товара на складе формирует заказ с меткой "отложен" | При импорте заявки, в случае отсутствия товара на складе, система формирует заказ со специальной меткой и размещает его в специальном реестре  | Логистика                     |
| 5ПродКл-5  | Процесс формирования заявки на основании заказа в соответствии с договором клиента | Разработать механизм копирования в Заявку строк из Заказа и Договора   | Шаблон файла импорта должен содержать номер договора, по которому формируется заявка   | Логистика                     |
| 5ПродКл-6  | Резервирование товара  | Разработать функционал, позволяющий резервировать заказанный товар   | Резервирование товара может происходить в системе двумя способами: автоматически и вручную   | Логистика                     |
| 5ПродКл-7  | Проверка кредитного лимита и дебиторской задолженности                             | Разработать алгоритм проверки условий  | При импорте заявки клиента системы проверяет кредитный лимит клиента. В случае превышения кредитного лимита система выдает сообщение о превышении кредитного лимита и блокирует дальнейшую обработку | Торговля                      |
| 5ПродКл-8  | Подбор номенклатурных единиц   | Автоматизации не подлежит  |  | Функциональность не требуется |
| 5ПродКл-10 | Формирование упаковочных листов  | Разработать функционал, позволяющий резервировать заказанный товар   |  | Логистика, управление складом |
| 5ПродКл-11 | Формирование счета, расходной накладной, счета-фактуры                             | Разработать электронную и печатную форму документов  |  | Торговля                      |
| 5ПродКл-12 | Отгрузка, списание медикаментов  | Разработать взаимосвязь с финансовым блоком  | Формирование проводки по выбытию товара со склада  | Логистика, торговля           |

## 7. Практическое занятие 4. Разработка глоссария

Выявить основную терминологию. Основные источники: результаты интервью, литературные источники, рекомендованные преподавателем, интернет. Составить глоссарий. Для составления глоссария целесообразно разбить проблемную область на подобласти.

Выполнить работу согласно образцу.

### Глоссарий

#### Введение

#### Цель

Глоссарий содержит описания терминов, используемых при проектировании информационной системы диспетчеризации типографии. Определяются основные понятия, непосредственно связанные с планированием и диспетчированием заказов.

#### Контекст

Глоссарий создан в рамках проекта автоматизации типографии «Печатник»

#### Ссылки

Сопутствующая информация представлена в документах [Диспетчеризация полиграфического производства.doc](#); [01-Vision.doc](#).

#### Определения

Понятия, используемые при описании исходной информации

#### Заказ

**Заказ** — заявленная заказчиком потребность в изготовлении продукции. Характеризуется свойствами (тираж, бумага, размеры и т.п.). Ассоциировано с совокупностью **работ**, упорядоченных в порядке исполнения.

#### Работа

**Работа** — единица предварительного планирования. Представляет собой работу на конкретном оборудовании (**ресурсе**) над одной единицей продукции (заказом). В задачах планирования и диспетчеризации не анализируется разбиение работы на технологические операции.

Работа является реализацией некоторого типа работ и относится к конкретному заказу.

#### Ресурс



**Ресурс** – единица оборудования, либо исполнитель, планируемая диспетчером.

Задание (элемент плана)

**Задание** – атомарная единица планирования диспетчером. В процессе предварительного планирования каждой работе соответствует своё задание (отношение 1:1). При планировании диспетчером в случае необходимости работы разбиваются на элементы – элементарные задания (например, длительность работы превышает длительность смены, продукция поступает порциями с интервалом и т.п.). Задание ассоциируется с одной единицей оборудования и одним или более **работниками**, бригадой. Один из работников является ответственным исполнителем.

Задание не может превышать по продолжительности рабочую смену.

Работник

**Работник** – атомарный человеческий ресурс при планировании. Работник имеет должность в соответствие со **штатным расписанием**. Согласно его должности, работнику делегируются полномочия по исполнению тех или иных работ.

Штатное расписание

**Штатное расписание** – не персонифицированный перечень должностей с указанием количества для каждой из них. Для каждой должности определяется перечень работ, которые должен выполнять работник, занимающий данную должность.

Бригада

**Бригада** – совокупность работников, работающих совместно в одну смену и в одном цехе. Диспетчером планируется объем и сроки выполнения работ целиком на бригаду.

Участок

**Цех** – группировка ресурсов (оборудования и персонала).

Смена

**Смена** – временной интервал в течение суток. Как правило, различают три 8-часовые, две двенадцатичасовые смены и работу «в день», например – с 9 до 18.

Расписание смены

**Расписание** – график чередования смен при работе бригады.

Понятия, используемые при планировании

Статус работы

**Статус работы** – состояние работы с точки зрения диспетчеризации. В первом приближении будем различать следующие статусы: *«работа заказа, принятого к исполнению»*, *«работа спланирована»*, *«работа частично выполнена»*, *«работа выполнена»*.

Допустимый интервал.

**Допустимый интервал** – сроки, в которые может быть выполнена (запланирована) работа, не нарушая технологической последовательности и сроков выполнения заказа.

Критический срок исполнения работ.

Срок начала работы в плане, перенос работы позже которого приводит к срыву сроков изготовления заказа исходя из технологических ограничений.

Коллизия

**Коллизия** – противоречивая информация в плане, приводящая либо к невозможности исполнения плана, либо к нарушению обязательств перед заказчиком.

Коллизии возможны в случаях:

два задания по обработке одной и той же единицы продукции имеют совмещение на временной шкале;

два задания по обработке различных единиц продукции, использующих один и тот же ресурс, имеют совмещение на временной шкале;

работа заканчивается после срока сдачи заказа.

Используемые документы

Предварительный план изготовления

**Предварительный план изготовления** – план, содержащий информацию обо всех работах, независимо от их состояния.

Общий план изготовления

**Общий план изготовления** – план, содержащий информацию только о работах в состоянии *«работа заказа, принятого к исполнению»*.

План-график исполнения заказа

**План-график исполнения заказа** – план, содержащий информацию (плановую и фактическую) обо всех работах по подготовке и изготовлении конкретного заказа.

Сменное задание

**Сменное задание** – выдержка из общего плана изготовления за определённую дату, смену и для определённого цеха.

## 8. Лабораторная работа 4. Подробное описание ключевых УС

Осуществить анализ ключевых УС. Основные критерии для отбора: критичная важность для пользователя (инвестора); УС, как "архитектурообразующий" компонент; УС, позволяющий снизить риски. Количественные критерии: доля ключевых УС, согласно правилу Парето, составляет порядка 20%, но не более 30%.

### **Цель работы**

- Необходимо специфицировать ключевую функциональность АИС.

### **Порядок выполнения работы**

– Осуществить анализ ключевых УС. Основные критерии для отбора: критичная важность для пользователя (инвестора); УС, как "архитектурообразующий" компонент; УС, позволяющий снизить риски. Количественные критерии: доля ключевых УС, согласно правилу Парето, составляет порядка 20%, но не более 30%.

– Описать основной поток событий ключевых УС. Здесь и далее использовать шаблон описания на основе методологии RUP

- Описать альтернативные сценарии ключевых УС.
- Оформить работу.
- Осуществить защиту работы.

### **Требования к оформлению результатов**

– Результаты должны быть представлены в документе MS Word; структура и форматирование должны быть взяты в примере оформления работы

- Необходимые технические средства
- Компьютеры для членов рабочей группы; MS Word.

### **Пример оформления работы**

#### **Описание ключевых прецедентов**

#### **Поиск ключевых вариантов использования**

Анализ сформулированных вариантов использования показал, что с точки зрения потенциальных рисков и архитектурной значимости наиболее существенными являются прецеденты, связанные с работой менеджера и диспетчера.

Для дальнейшей детализации выбраны три прецедента:

M1. Регистрация заказа;

D1. Планирование нового заказа;

D3. Планирование срочного заказа.

Прецедент D1: планирование нового заказа

Планирование нового заказа

Краткое описание

Диспетчер размещает вновь поступивший от менеджера заказ в план в «хвост» очереди.

*Действующие лица этого прецедента – Диспетчер.*

Поток событий

Прецедент начинается, когда Диспетчер выбирает деятельность “планировать новый заказ” из «Главной формы» АРМ «Диспетчер».

Базовый поток – Планирование нового заказа

Диспетчер выбирает «планировать новый заказ».

Система отображает список новых заказов, подлежащих планированию.

Диспетчер выбирает из предложенного списка заказ, который он желает запланировать.

Система определяет, что статус заказа – «Обычный».

Система отображает список работ заказа, отсортированных по очередности исполнения с указанием времени исполнения.

Система отображает список совместимых ресурсов. Ресурс отображается в том случае, если в заказе есть хотя бы одна работа, для которой он допустим.

Система отображает для каждого из ресурсов линейки планирования, состоящие из свободных и занятых временных интервалов на шкале времени.

Диспетчер выбирает работу заказа.

Система ограничивает набор доступных ресурсов, «затеняя» несовместимые.

Диспетчер находит на шкале одного из доступных ресурсов интервал необходимого размера и размещает (drag and drop) туда работу заказа.

Система делает соответствующие отметки в базе данных.

ПП. 7-10 повторяются, пока все работы заказа не будут размещены.

Система удаляет заказ из списка вновь поступивших.

Альтернативные потоки

Планирование по частям

Если при выполнении п. 0 основного потока событий Диспетчеру не удалось обнаружить интервал необходимого раздела, то

Диспетчер выбирает «планировать по частям».

Диспетчер находит на шкале одного из доступных ресурсов интервал произвольного размера и размещает (drag and drop) туда работу заказа.

Система разбивает работу на интервалы и размещает её на свободные позиции выбранного ресурса.

Переход к п. 0 основного потока событий.

Планирование заказа в срок невозможно

Если Диспетчер обнаружил, что он не может запланировать заказ с соблюдением зафиксированного в заказе срока, то

Диспетчер выбирает «отменить планирование».

Система отправляет уведомление Менеджеру «Заказ №... не может быть спланирован с соблюдением оговоренного с заказчиком срока».

Специальные требования

Время планирования одного заказа не должно превышать 3 минут.

Предусловия

Регистрация

Перед тем как начинается этот прецедент, Диспетчер зарегистрирован в системе.

Постусловия

При успешном окончании прецедента Диспетчер составляет план, гарантирующий исполнение заказа в срок. При неуспешном – Диспетчер делегирует ответственность за соблюдение сроков исполнения заказа Менеджеру (менеджер указал некорректный срок и должен его откорректировать).

Точки расширения

Если при выполнении выясняется, что заказ имеет статус «Срочный», Система переходит к выполнению расширяющего прецедента «Прецедент D3. Планирование срочного заказа»

Прецедент M1. Регистрация заказа

Регистрация заказа

Краткое описание

Поток событий

Базовый поток – Регистрация заказа

Альтернативные потоки

Специальные требования

Предусловия

Постусловия

Точки расширения

Прецедент D3. Планирование срочного заказа

Планирование срочного заказа

Краткое описание

Поток событий

Базовый поток – Планирование срочного заказа

Альтернативные потоки

Специальные требования

Предусловия

Постусловия

Точки расширения

## 9. Практическое занятие 5. Основы работы с CASE средством AllFusion Process Modeler

AllFusion Process Modeler (далее BPwin) — CASE-средство для моделирования бизнес-процессов, позволяющая создавать диаграммы в нотации IDEF0, IDEF3, DFD. В процессе моделирования BPwin позволяет переключиться с нотации IDEF0 на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. BPwin поддерживает функционально-стоимостной анализ (ABC).

Работа с программой начинается с создания новой модели, для которой нужно указать имя и тип (рис.1).

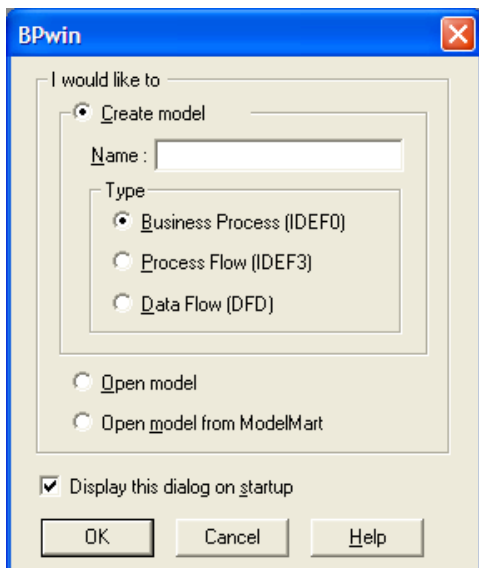


Рисунок 1. Создание новой модели

От выбора типа модели зависит в каких нотациях можно производить декомпозицию работ. Так, если выбрать тип *Business Process (IDEF0)*, то в созданной модели можно производить декомпозицию работ в нотациях IDEF0, IDEF3 и DFD; если выбран тип *Data Flow (DFD)* — в нотациях DFD и IDEF3; если же выбран тип *Process Flow (IDEF3)* — то только в нотации IDEF3.

После ввода имени модели и выбора ее типа BPWin сразу предложит задать параметры модели (рис. 2):

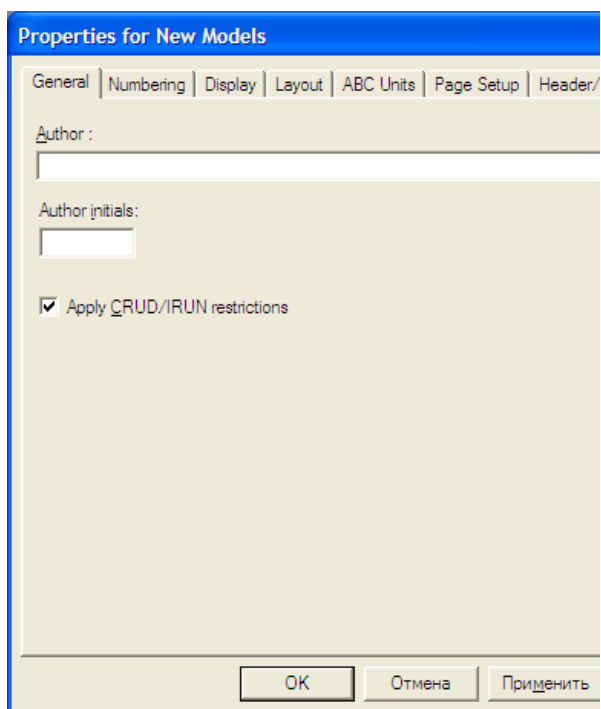


Рисунок 2. Окно задания свойств модели

- *General*— автор модели и его инициалы;
- *Numbering* — формат нумерации работ и диаграмм и порядок ее отображения на диаграммах;
- *Display* — список элементов отображения на диаграммах;
- *Layout* — параметры расположения;
- *ABC Units* — единицы функционально-стоимостного анализа;
- *Page Setup* — параметры страницы;
- *Header/Footer* — параметры верхнего и нижнего колонтитула.

После задания свойств модели появляется главное окно программы (рис. 3), состоящее из трех основных частей:



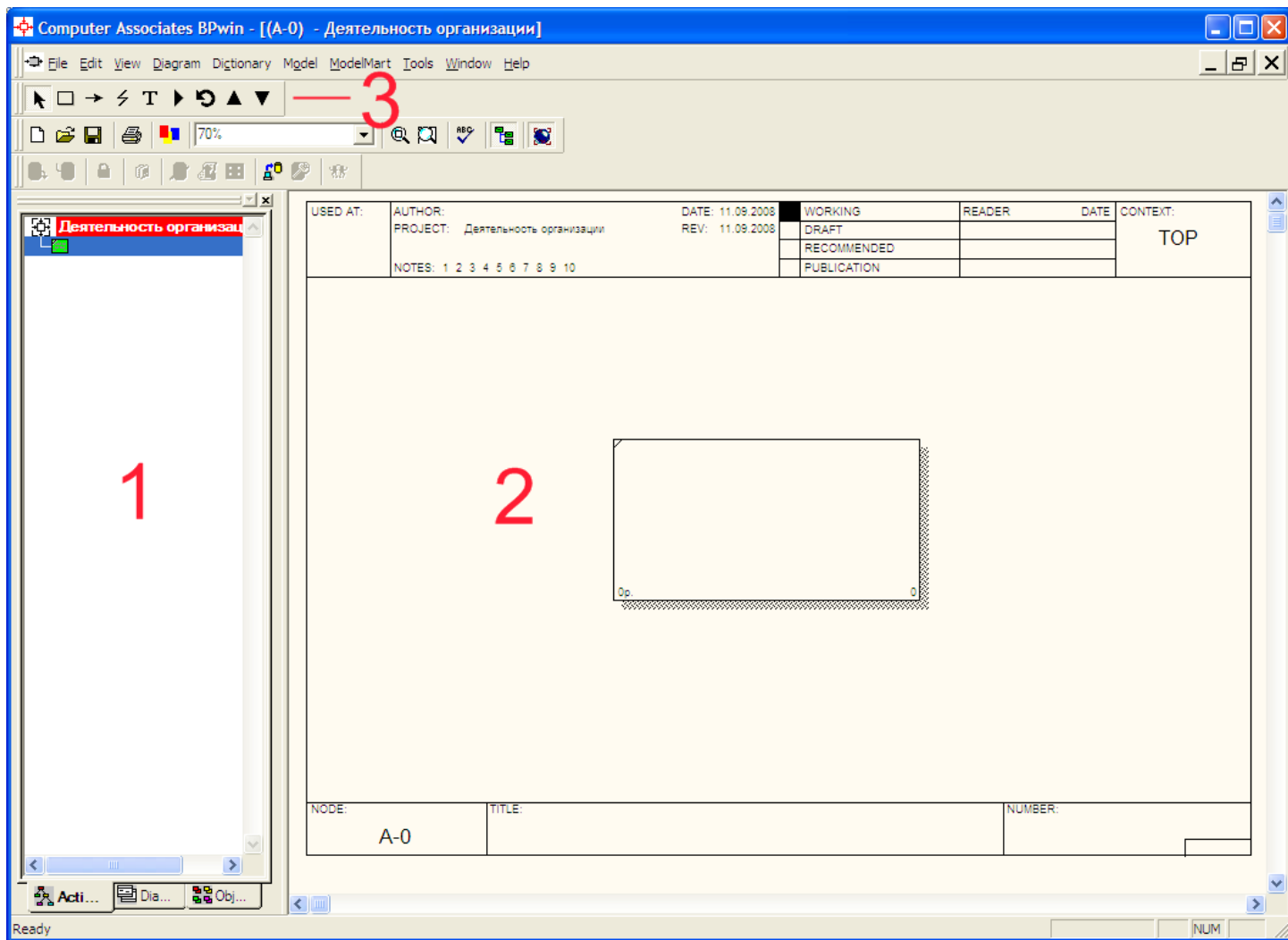


Рисунок 3. Главное окно программы

- 1 - обозреватель модели (Model Explorer) — отображает структуру модели (имеющиеся диаграммы и их иерархию);
- 2 - основная часть — в ней отображаются диаграммы, с которыми ведется работа;
- 3 - панели инструментов, из которых наибольший интерес представляет панель инструментов *Model Toolbox*.

**Примечание.** В созданной модели с настройками по умолчанию некорректно отображаются русские символы. Чтобы устранить этот недостаток, необходимо подкорректировать используемые в модели шрифты. Для этого в меню *Model -> Default Fonts* необходимо последовательно пройти по всем пунктам (рис. 4), выбрать в выпадающем списке *Script* значение *кириллический* и поставить галочку *Change all occurrences* (рис. 5).

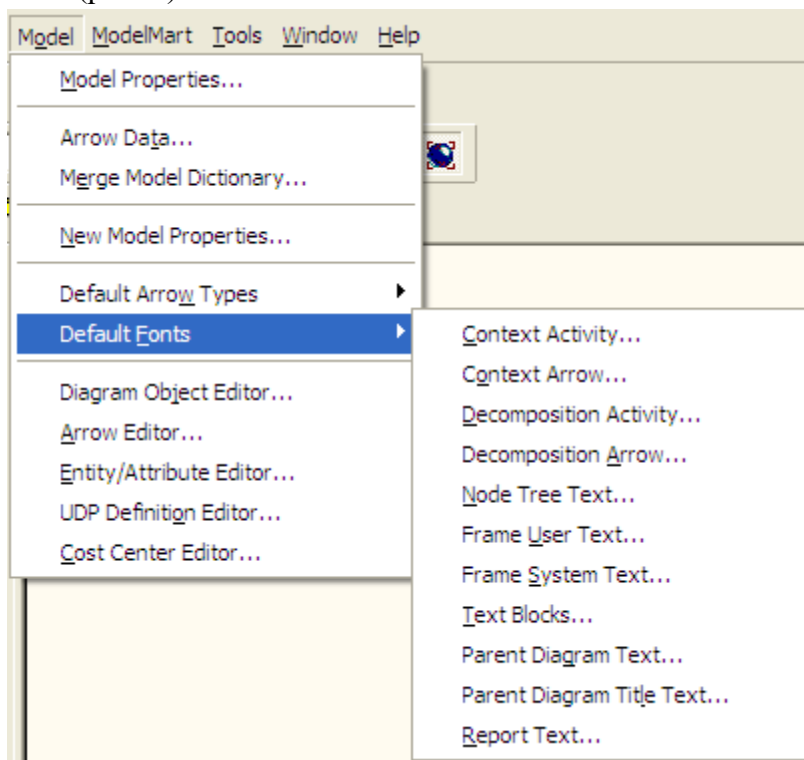


Рисунок 4. Пункты меню, отвечающие за настройки шрифта

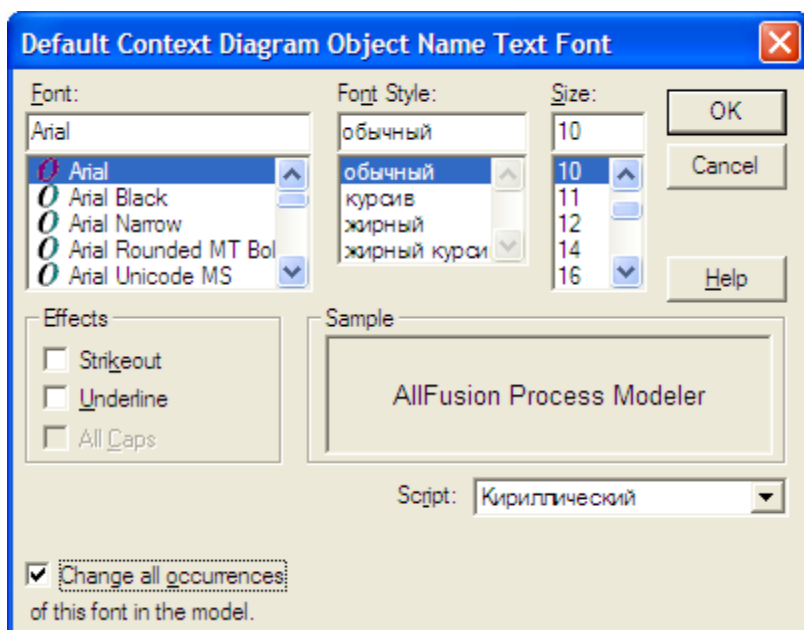



Рисунок 5. Параметры шрифта

**Панель инструментов *Model Toolbox*.**

Данная панель инструментов отвечает за создание разнообразных графических элементов модели. В зависимости от типа текущей диаграммы набор кнопок на ней меняется.

Таблица 1 - Вид и назначение кнопок *Model Toolbox*

| Вид кнопки  | Название кнопки           | Назначение кнопки   |
|---|---------------------------|---|
|    | Pointer Tool              | Превращает курсор в стрелку указателя для того, чтобы можно было выделять объекты             |
|  - IDEF0<br> - DFD<br> - IDEF3 | Activity Box Tool         | Добавление на диаграмму новой работы  |
|    | Precedence Arrow Tool     | Добавление на диаграмму новой стрелки   |
|    | Squiggle Tool             | Связывание названия стрелки с самой стрелкой  |
|    | Text Tool                 | Добавление на диаграмму текста  |
|    | Diagram Dictionary Editor | Вызов окна менеджера диаграмм для просмотра имеющихся диаграмм по типам и переход к выбранной |
|    | Go to Sibling Diagram     | Переход между стандартной диаграммой, деревом узлов и FEO диаграммой                          |
|    | Go to Parent Diagram      | Переход к родительской диаграмме  |
|    | Go to Child Diagram       | Переход к дочерней диаграмме  |
|  - DFD   | External Reference Tool   | Добавление на диаграмму внешней сущности  |
|  - DFD   | Data store Tool           | Добавление на диаграмму хранилища данных  |
|  - IDEF3   | Junction Tool             | Добавление на диаграмму перекрестка   |
|  - IDEF3   | Referent Tool             | Добавление на диаграмму объекта ссылки  |

Созданная модель уже содержит контекстную диаграмму с единственной работой ("черный ящик") в той нотации, которая была выбрана на этапе создания модели. Теперь необходимо дать этой работе название и при необходимости задать ее свойства. Для этого нужно вызвать окно свойств работы, дважды щелкнуть по ней мышью (рис. 6).

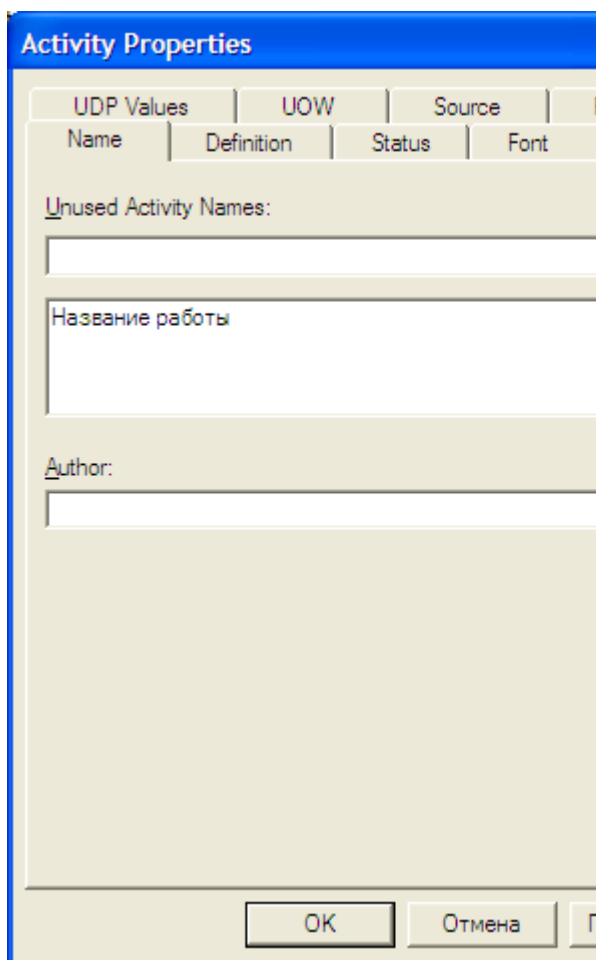


Рисунок 6. Окно свойств работы

Далее необходимо разместить на диаграмме стрелки. Для этого следует нажать на *Model Toolbox* кнопку *Precedence Arrow Tool* (курсор примет форму крестика со стрелкой), щелкнуть по тому месту, откуда стрелка должна выходить и затем щелкнуть по тому месту, куда стрелка должна заходить (BPwin подсветит эти места при наведении на них курсора). Для задания названия стрелки нужно нажать на *Model Toolbox* кнопку *Pointer Tool* и затем дважды щелкнуть по стрелке. В появившемся окне *Arrow Properties* название работы вводится в поле *Arrow Name* или выбирается из списка имеющихся названий стрелок.

После размещения стрелок на диаграмме можно проводить декомпозицию ее работ. Для этого следует нажать на *Model Toolbox* кнопку *Go to Child Diagram* и затем щелкнуть по работе, которую нужно декомпозировать. Появится окно, в котором необходимо выбрать в какой нотации проводить декомпозицию и количество дочерних работ (рис. 7).

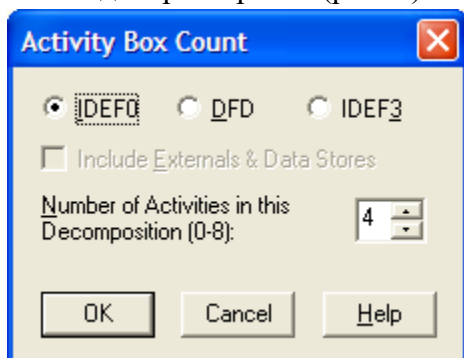


Рисунок 7. Создание дочерней диаграммы

После создания дочерней диаграммы BPwin автоматически создаст указанное число работ и разместит граничные стрелки по краям диаграммы. Далее следует связать граничные стрелки со входами работ (при необходимости можно добавить новые граничные стрелки) и связать работы между собой. Дальнейшая декомпозиция работ проводится аналогичным образом.

## Начало моделирования. Построение контекстной диаграммы в нотации IDEF0

### Цель работы:

- кратко описать выбранную предметную область (чем занимается предприятие, какие основные процессы в нем происходят)
- определить контекст моделирования
- построить контекстную диаграмму в нотации IDEF0

Вариант индивидуального задания необходимо согласовать с преподавателем. Все диаграммы в нотациях IDEF0, IDEF3 и DFD строить в рамках одной модели в AllFusion Process Modeler. Целью данной и большинства последующих работ является моделирование деятельности выбранного предприятия. Для этого будут применяться методологии:

- IDEF0 - методология функционального моделирования
- IDEF3 - методология описания процессов
- DFD - методология моделирования потоков данных
- IDEF1X - методология моделирования данных

Диаграммы в первых трех методологиях будут создаваться с помощью CASE-средства AllFusion Process Modeler, IDEF1X - с помощью AllFusion ERwin Data Modeler. Каждая диаграмма в нотациях IDEF0, IDEF3, DFD предназначена для описания одного или нескольких бизнес-процессов. **Бизнес-процесс** - это устойчивая, целенаправленная совокупность взаимосвязанных видов деятельности (последовательность работ), которая по определенной технологии преобразует входы в выходы, представляющие ценность для потребителя. Результатом моделирования бизнес-процессов является модель бизнес-процессов, которая относится к одному из трех типов:

- модель AS-IS (как есть) - модель текущей организации бизнес-процессов предприятия
- модель TO-BE (как будет) - модель идеальной организации бизнес-процессов
- модель SHOULD-BE (как должно бы быть) - идеализированная модель, не отражающая реальную организацию бизнес-процессов предприятия

В лабораторных работах будет создаваться модель AS-IS. Перед началом построения диаграмм необходимо изучить выбранную предметную область. В этой и последующих работах в качестве предметной области будет выступать вымышленное предприятие по сборке и продаже настольных компьютеров и ноутбуков. Компания не производит комплектующие самостоятельно, а только собирает и тестирует компьютеры. Основные процедуры в компании:

- продавцы принимают заказы клиентов;
- сотрудники группируют заказы по типам компьютеров;
- сотрудники собирают и тестируют компьютеры;
- сотрудники упаковывают компьютеры согласно заказам;
- кладовщик отгружает клиентам заказы
- снабженцы заказывают и доставляют комплектующие, необходимые для сборки.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам. Построение модели какой-либо системы в методологии IDEF0 начинается с определения **контекста моделирования**, который включает в себя субъекта моделирования, цель моделирования и точку зрения на модель.

Под **субъектом** понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, необходимо определить, что в дальнейшем будет рассматривать как компоненты системы, а что как внешнее воздействие.

**Цель моделирования**. Модель не может быть построена без четко сформулированной цели. Цель должна отвечать на следующие вопросы:

- Почему этот процесс должен быть смоделирован?
- Что должна показывать модель?
- Что может получить читатель?

**Точка зрения.** Не смотря на то, что при построении модели учитываются мнения различных людей, модель должна строиться с единой точки зрения. Точку зрения можно представить как взгляд человека, который видит систему в нужном для моделирования аспекте. Точка зрения должна соответствовать цели моделирования. В течении моделирования важно оставаться на выбранной точке зрения. В данной работе субъектом будет выступать само предприятие, а именно процессы, происходящие внутри него; цель моделирования - воспроизвести бизнес-процессы, происходящие на предприятии (модель AS-IS); точка зрения - с позиции директора как лица, знающего структуру предприятия в целом. После определения контекста моделирования можно приступить к построению контекстной диаграммы (называемой еще "черным ящиком"). Данный тип диаграммы позволяет показать, что подается на вход работы и что является результатом работы, без детализации ее составляющих. Данная диаграмма содержит только одну работу, которая будет представлять всю деятельность предприятия в целом (рис.1).

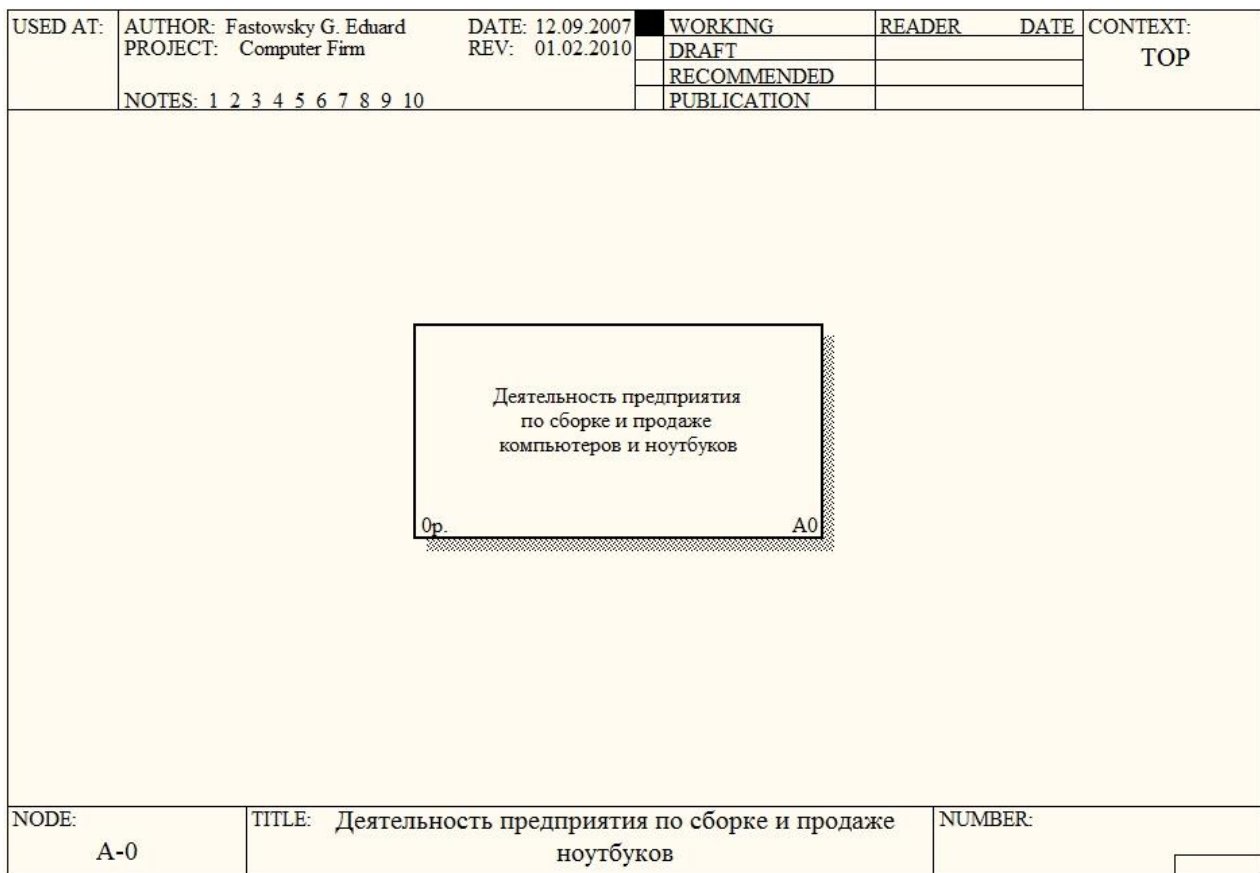


Рисунок 1. Контекстная диаграмма

Любая IDEF0 диаграмма состоит из прямоугольников, называемых работами (activity), и стрелок (arrow). Работа представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждой работы должно быть выражено отглагольным существительным (например, "Изготовление детали", "Оформление заказа" и т.д.). Каждая из четырех сторон прямоугольника имеет свое определенное значение (рис.2):

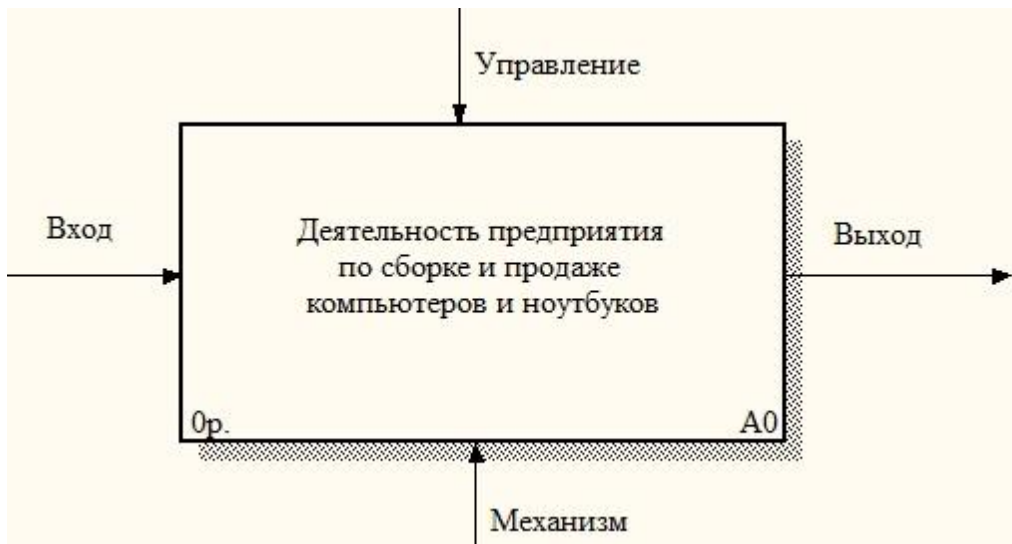


Рисунок 2. Работа в IDEF0

- Вход – это потребляемая или изменяемая работой информация или материал
- Выход – информация или материал, которые производятся работой
- Управление – процедуры, правила, стратегии или стандарты, которыми руководствуется работа

Механизмы – ресурсы, которые выполняют работу (например, сотрудники, оборудование, устройства и т.д.)

Для рассматриваемого предприятия *входными стрелками* будут:

- Заказы клиентов - список компьютеров и их конфигурация, которые клиент желает приобрести
- Комплектующие от поставщиков - комплектующие, полученные от поставщиков, из которых собираются компьютеры и ноутбуки

*Выходные стрелки:*

- Готовая продукция - собранные компьютеры и ноутбуки
- Заказы поставщикам - список комплектующих, которые предприятие закупает у поставщиков
- Оплата за комплектующие - деньги поставщикам за комплектующие
- Маркетинговые материалы - прайс-листы, рекламки и т.п.

*Стрелки управления:*

- Законодательство - различные законодательные документы, которыми руководствуется предприятие в процессе своей деятельности
- Правила и процедуры - различные правила и процедуры, которыми руководствуется предприятие в процессе своей деятельности (например, правила сборки и тестирования компьютеров, процедура общения с клиентами и т.п.)

*Стрелки механизмов:*

- Бухгалтерская система
- Персонал

Итоговая контекстная диаграмма имеет вид (рис.3):

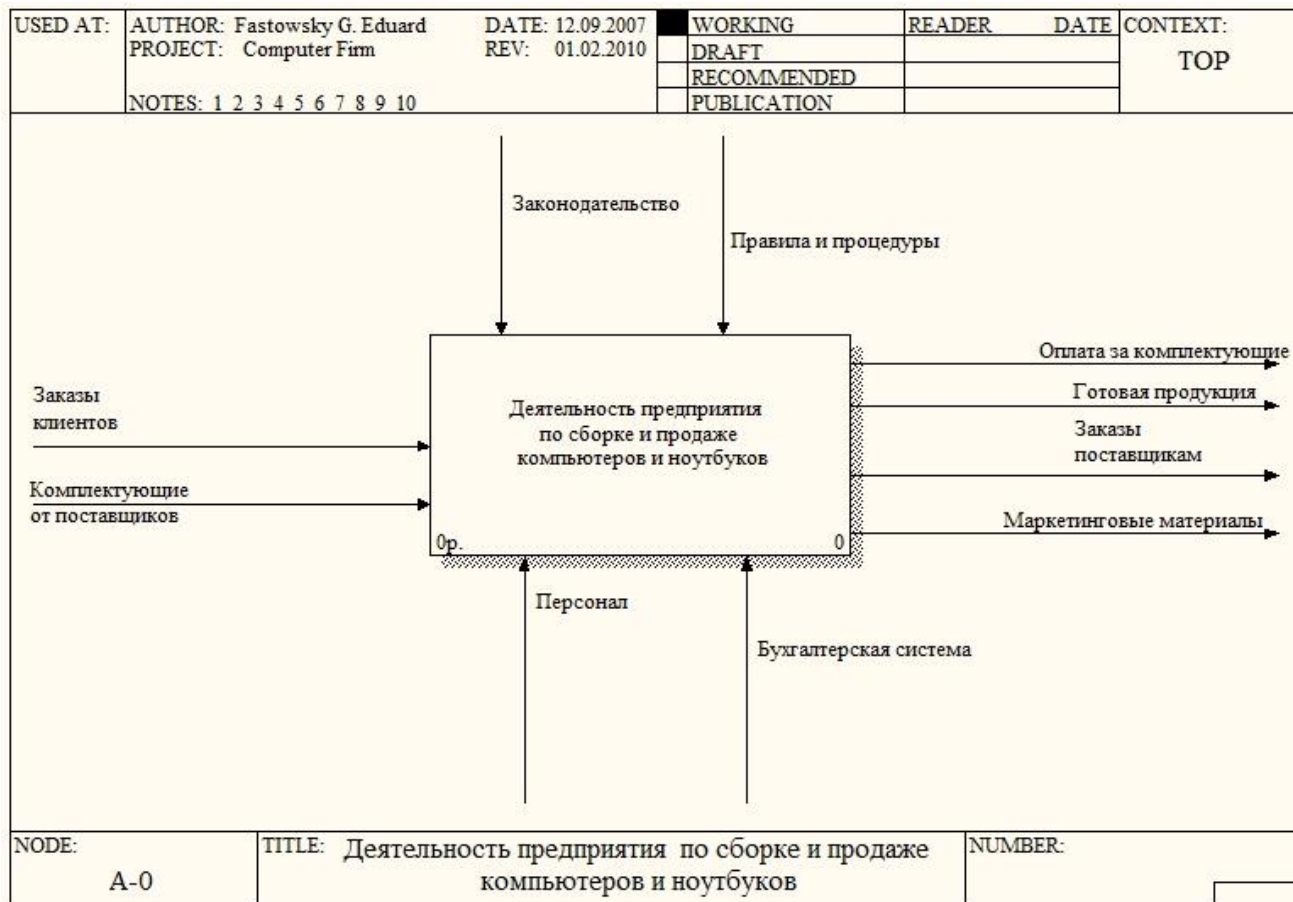


Рисунок 3. Итоговая контекстная диаграмма

**Содержание отчета:**

- вариант индивидуального задания
- краткое описание выбранной предметной области (чем предприятие занимается, как функционирует)
- описание контекста моделирования
- контекстная диаграмма



## Построение диаграммы декомпозиции второго уровня в нотации IDEF0

### Цель работы:

- построить диаграмму декомпозиции второго уровня в нотации IDEF0

В предыдущей работе была построена контекстная диаграмма, содержащая только одну работу, которая описывает деятельность предприятия в целом, без детализации составляющих этой работы. В данной работе будет построены диаграммы декомпозиции первого и второго уровней в нотации IDEF.

**Декомпозиция** - это разделение сложного объекта, системы, задачи на составные части, элементы. С помощью диаграммы декомпозиции первого уровня покажем, из каких более мелких работ состоит работа "Деятельность предприятия по сборке и продаже компьютеров и ноутбуков". В данной работе были выделены следующие дочерние работы:

|                                   |   |
|-----------------------------------|---|
| Управление                        | Данная работа включает в себя общее управление предприятием, финансами, кадрами, бухгалтерию и т.п. |
| Продажи и маркетинг               | Работа с клиентами, презентации, выставки, реклама, маркетинговые исследования и т.д.               |
| Сборка и тестирование компьютеров | Сборка и тестирование настольных компьютеров и ноутбуков  |
| Отгрузка и снабжение              | Снабжение предприятия необходимыми комплектующими, хранение и отгрузка готовой продукции            |

После создания дочерней диаграммы первым действием соединим граничные стрелки с работами. Стрелку "Заказы клиентов" соединим с работой "Продажи и маркетинг", стрелку "Комплектующие от поставщиков" - с "Отгрузка и снабжение". Выходом работы "Управление" будет "Оплата за комплектующие", выходом "Продажи и маркетинг" - "Маркетинговые материалы". Стрелки "Заказы поставщикам" и "Готовая продукция" - выход работы "Отгрузка и снабжение".

Стрелка "Персонал" будет являться входом механизма всех четырех работ, а стрелка "Бухгалтерская система" - работ "Продажи и маркетинг" и "Отгрузка и получение". Стрелка "Правила и процедуры" будет входом управления всех четырех работ. Любую ветвь стрелки также можно декомпонировать и дать ей свое название. Покажем это на примере ветки стрелки "Бухгалтерская система" для работы "Продажи и маркетинг". Назовем ее "Система оформления заказа". В AllFusion Process Modeler для более четкого указана какая-то стрелке относятся существуют несколько механизмов, одним из которых является **Squiggle** - стрелка в виде молнии, соединяющая название со стрелкой. Воспользуемся им для, для чего щелкнем правой кнопкой по названию стрелки и выберем в выпадающем меню соответствующий пункт.

На данном этапе построения диаграммы выяснилось, что мы не учли такой важный фактор, как деньги, которые клиенты дают за готовую продукцию. Деньги клиентов - это вход работы "Деятельность предприятия по сборке и продаже компьютеров и ноутбуков". Добавим эту стрелку на диаграмму декомпозиции.

Если по каким-то причинам граничную стрелку дочерней диаграммы не следует показывать (например, на данной диаграмме она является несущественной, или чтоб не загромождать диаграмму), то ее можно просто удалить.

Удалим

стрелку "Законодательство".

Результат всех перечисленных действий показан на рис.1.

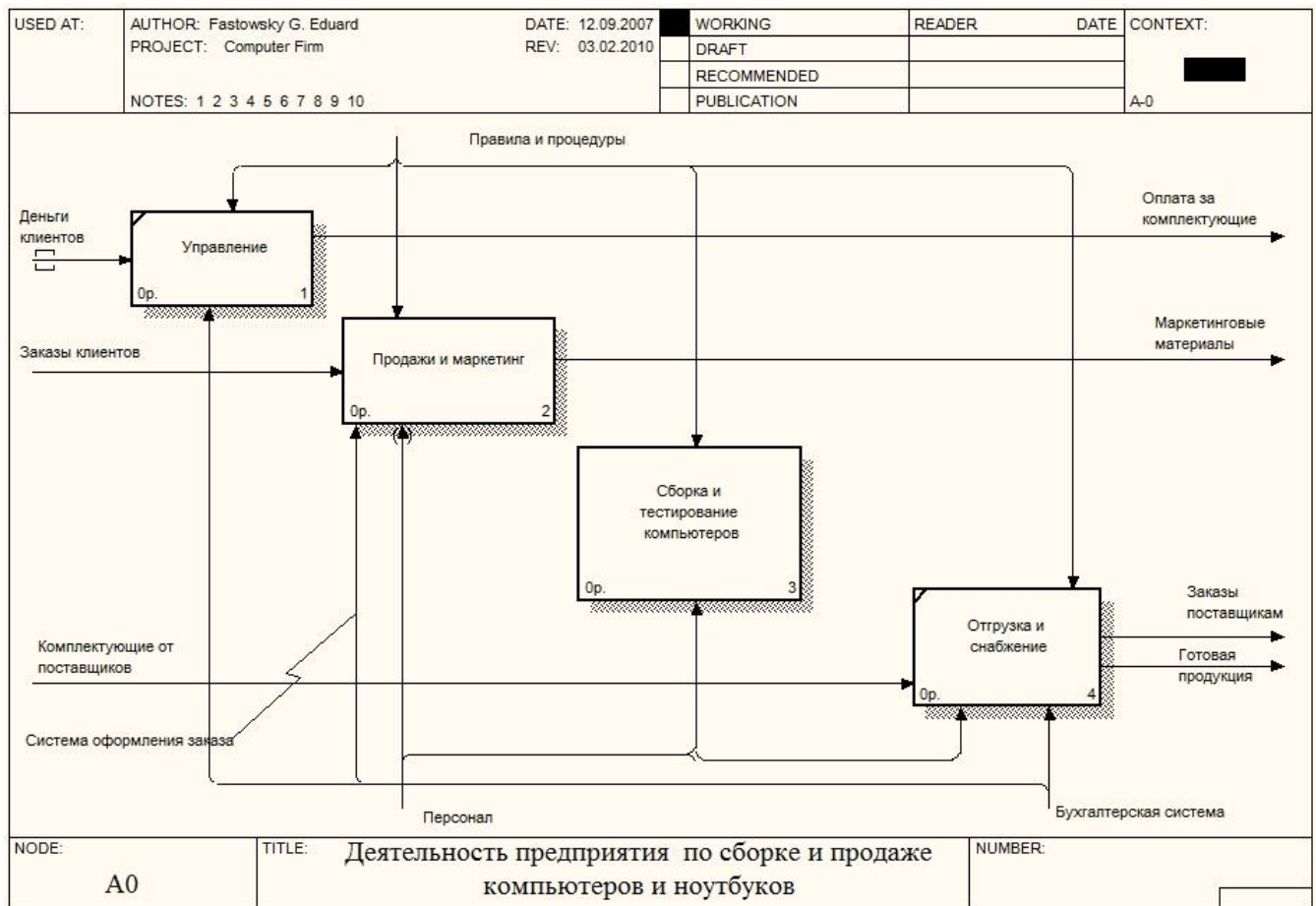


Рисунок 1

Если посмотреть на стрелку "Деньги клиентов" диаграммы декомпозиции и на стрелку "Законодательство" контекстной диаграммы, то видно, что они окружены небольшими квадратными скобками. Это означает, что данная граничная стрелка является новой на диаграмме и ее нет на дочерней диаграмме (как в случае со стрелкой "Законодательство"), или же данная стрелка является новой на дочерней диаграмме и ее нет на родительской (как в случае со стрелкой "Деньги клиентов"). От стрелок с квадратными скобками необходимо избавляться. Для этого есть два пути:

- добавить их на родительскую или дочернюю диаграмму, т.е. сделать граничной
- затоннелировать

Чтоб добавить такую стрелку на другую диаграмму или затоннелировать, нужно щелкнуть по квадратным скобкам правой кнопкой мыши и выбрать пункт меню "**Arrow Tunnel**". В появившемся окне следует выбрать один из двух вариантов: *Resolve it to border arrow* - сделать стрелку граничной, *Change it to resolved rounded tunnel* - затоннелировать стрелку. В данном случае мы решили обе стрелки затоннелировать (рис.2

и

рис.3).

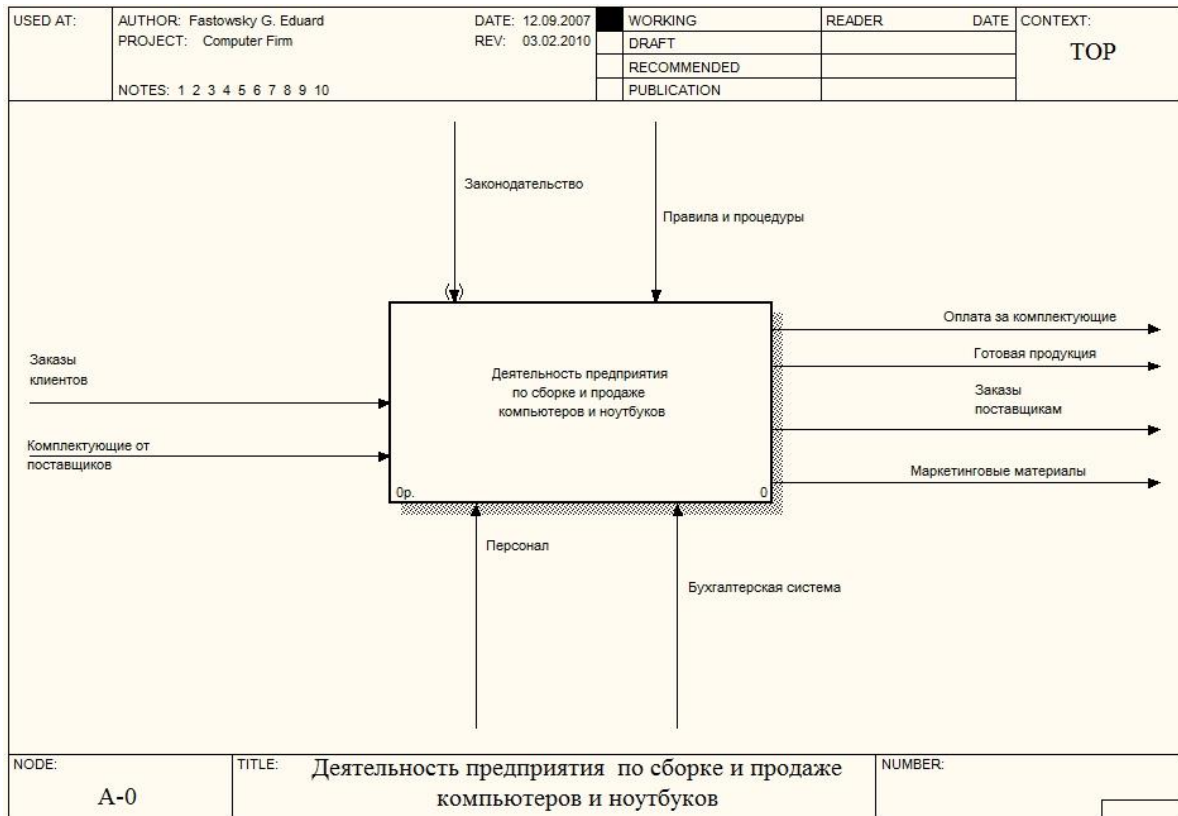


Рисунок 2. Контекстная диаграмма с затоннелированной граничной стрелкой

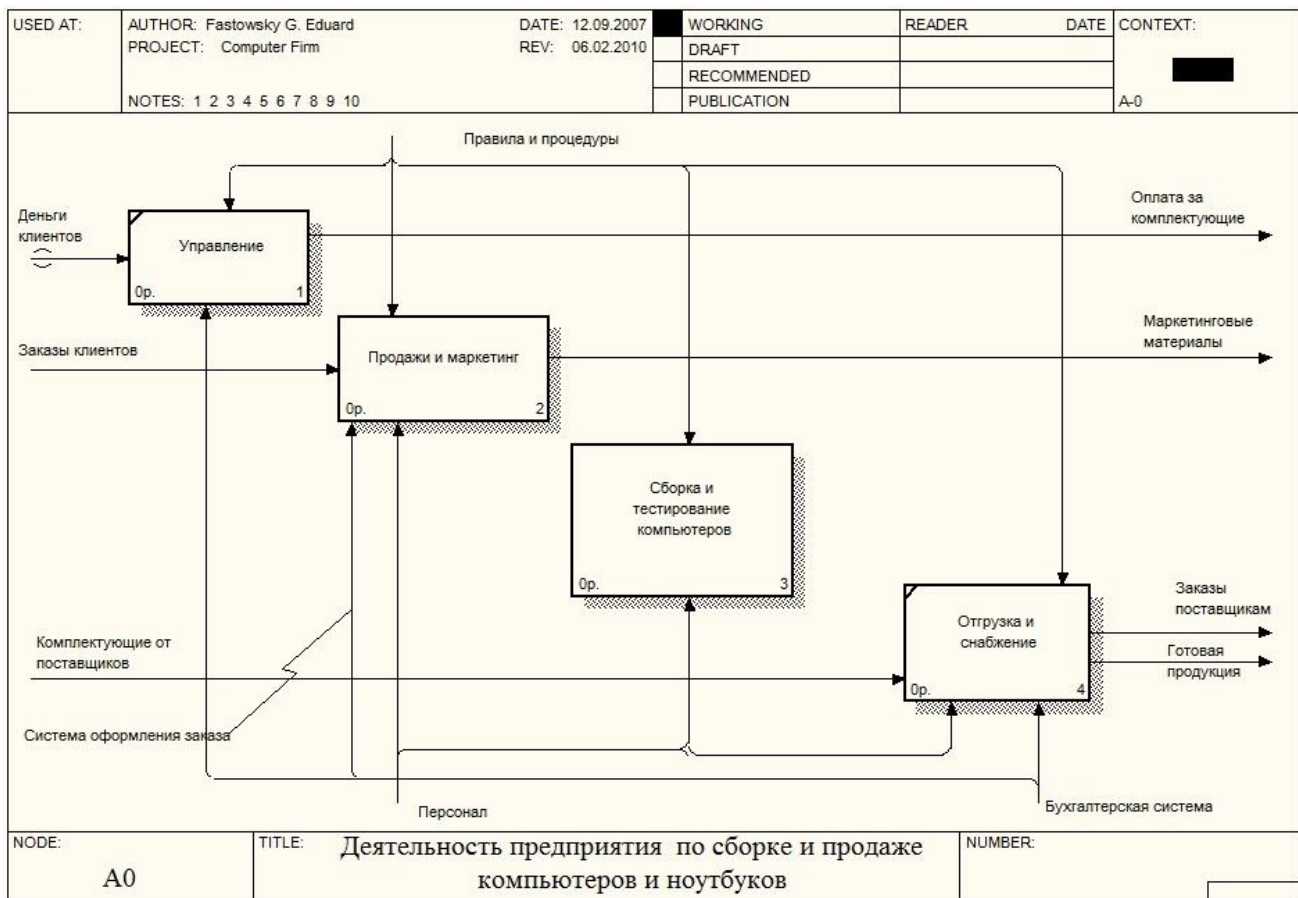


Рисунок 3. Диаграмма декомпозиции с затоннелированной граничной стрелкой

После соединения граничных стрелок с работами следующим шагом соединим работы между собой. Поскольку работа "Управление" включает в себя общее управление предприятием, то одним из ее результатов будет являться "Управляющая информация", поступающая на вход управления всех остальных работ.

Работа "Продажи и маркетинг" получает на входе заказы клиентов (т.е. количество компьютеров и их конфигурация), информацию о которых она передает работе "Сборка и тестирование компьютеров" в качестве управляющей информации. Работе "Сборка и тестирование" для своего функционирования необходимы комплектующие, которые она заказывает у работы "Отгрузка и снабжение" (выходная стрелка "Список необходимых комплектующих"). Собранные компьютеры она также передает работе "Отгрузка и снабжение" (выходная стрелка "Собранные компьютеры"). Информация о результатах сборки и тестирования необходима работе "Продажи и маркетинг" (выходная стрелка "Результаты сборки и тестирования"). Результатом работы "Отгрузка и снабжение" будут необходимые комплектующие, которые поступают на вход работы "Сборка и тестирование компьютеров". Управление любого предприятия должно знать, что происходит на предприятии, чем занимается каждое подразделение и каковы результаты их работы, т.е. любая работа в идеале должна отчитываться о результатах своей деятельности перед управлением. Создадим стрелки выходов работ "Продажи и маркетинг", "Сборка и тестирование компьютеров" и "Отгрузка и снабжение" и соединим их со входом управления работы "Управление". Результат соединения работ между собой показан на рисунке 3:

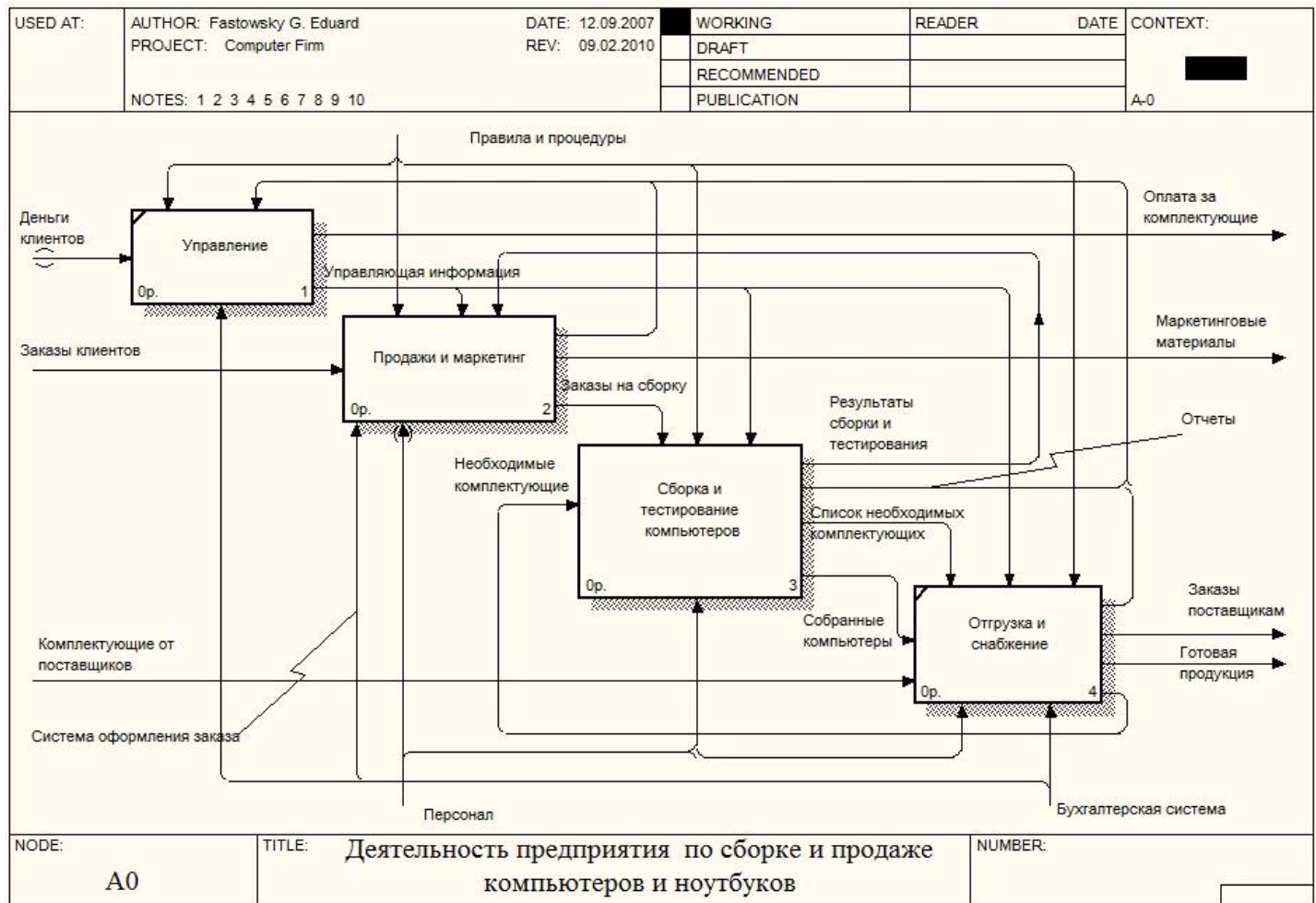


Рисунок 3

Если на диаграмме присутствует много работ и стрелок, то бывает затруднительно читать диаграмму. Для облегчения изучения диаграммы отдельные стрелки можно визуально выделить. Для зрительного выделения стрелки, соединяющей две работы, есть несколько механизмов:

- задать толщину стрелки
- поменять цвет стрелки
- добавить на стрелку дополнительные наконечники

Толщина и цвет стрелки задаются в окне свойств стрелки, вызываемое двойным щелчком по стрелке. Вкладка "Style" отвечает за стиль стрелки, в том числе и за ее толщину ("Thickness"), вкладка "Color" - за ее цвет. Для добавления на стрелку дополнительных наконечников следует щелкнуть правой кнопкой по

стрелке и выбрать пункт меню **"Extra Arrowhead"**.

Модифицируем диаграмму, визуально выделив некоторые стрелки. Итоговая диаграмма декомпозиции показана на рисунке 4:

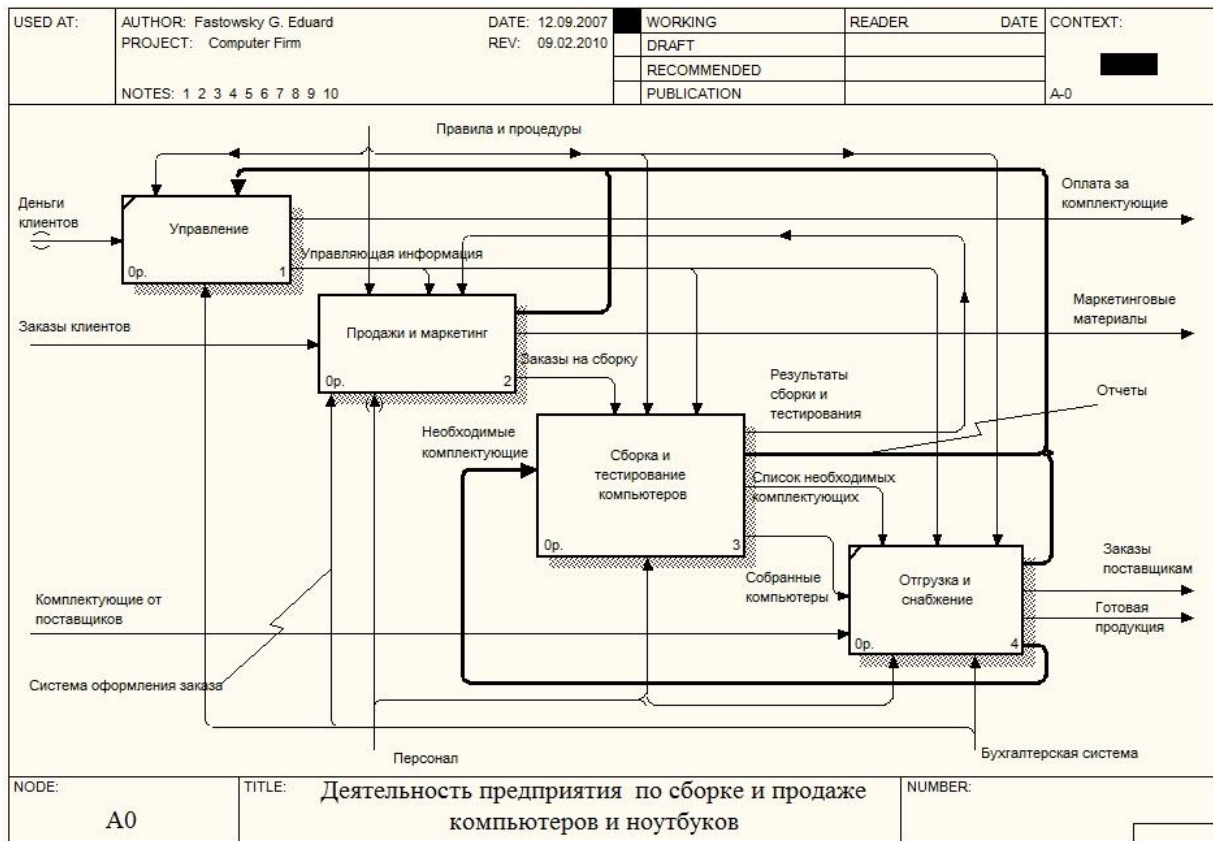


Рисунок 4. Итоговая диаграмма декомпозиции первого уровня

#### Содержание отчета:

- диаграмма декомпозиции второго уровня
- краткое описание каждой работы

## 10. Лабораторная работа 5 Работа с CASE средством AllFusion Process Modeler

### Цель работы:

- построить диаграмму декомпозиции следующего уровня в нотации IDEF0

В данной лабораторной работе построим еще одну диаграмму декомпозиции в нотации IDEF0 - декомпозицию работы *"Сборка и тестирование компьютеров"* диаграммы A0.

В результате проведения экспертизы получена следующая информация.

- Производственный отдел получает заказы клиентов от отдела продаж по мере их поступления.
- Диспетчер координирует работу сборщиков, сортирует заказы, группирует их и дает указание на отгрузку компьютеров, когда они готовы.
- Каждые 2 часа диспетчер группирует заказы - отдельно для настольных компьютеров и ноутбуков - и направляет на участок сборки.
- Сотрудники участка сборки собирают компьютеры согласно спецификациям заказа и инструкциям по сборке. Когда группа компьютеров, соответствующая группе заказов, собрана, она направляется на тестирование.
- Тестировщики тестируют каждый компьютер и в случае необходимости заменяют неисправные компоненты. Тестировщики направляют результаты тестирования диспетчеру, который на основании этой информации принимает решение о передаче компьютеров, соответствующих группе заказов, на отгрузку.

В данной работе мы выделили четыре дочерних работы: *"Отслеживание расписания и управление сборкой и тестированием"*, *"Сборка настольных компьютеров"*, *"Сборка ноутбуков"* и *"Тестирование компьютеров"*. Как и в предыдущей работе начнем с соединения граничных стрелок с работами.

Стрелка *"Необходимые комплектующие"* - это вход работ *"Сборка настольных компьютеров"* и *"Сборка ноутбуков"*.

Стрелки управления *"Управляющая информация"* и *"Заказы на сборку"* соединим с работой *"Отслеживание расписания и управление сборкой и тестированием"*, поскольку именно данная работа управляет всем процессом сборки и тестирования, а стрелку управления *"Правила и процедуры"* - с остальными тремя работами.

Персонал принимает участие во всех выделенных дочерних работах, поэтому заводим стрелку *"Персонал"* на вход механизма всех работ (при этом указав, что в первой работе участвует диспетчер, а в четвертой - тестировщик).

Список необходимых комплектующих - это один из результатов работ *"Сборка настольных компьютеров"* и *"Сборка ноутбуков"*. Результаты сборки и тестирования - это выходы работ *"Сборка настольных компьютеров"*, *"Сборка ноутбуков"* и *"Тестирование компьютеров"*. Компьютеры считаются собранными после того, как они успешно прошли тестирование, поэтому стрелка выхода *"Собранные компьютеры"* - выход работы *"Тестирование компьютеров"*. Различные отчеты формирует работа *"Отслеживание расписания и управление сборкой и тестированием"*.

Результат проделанных операций показан на рисунке 1:



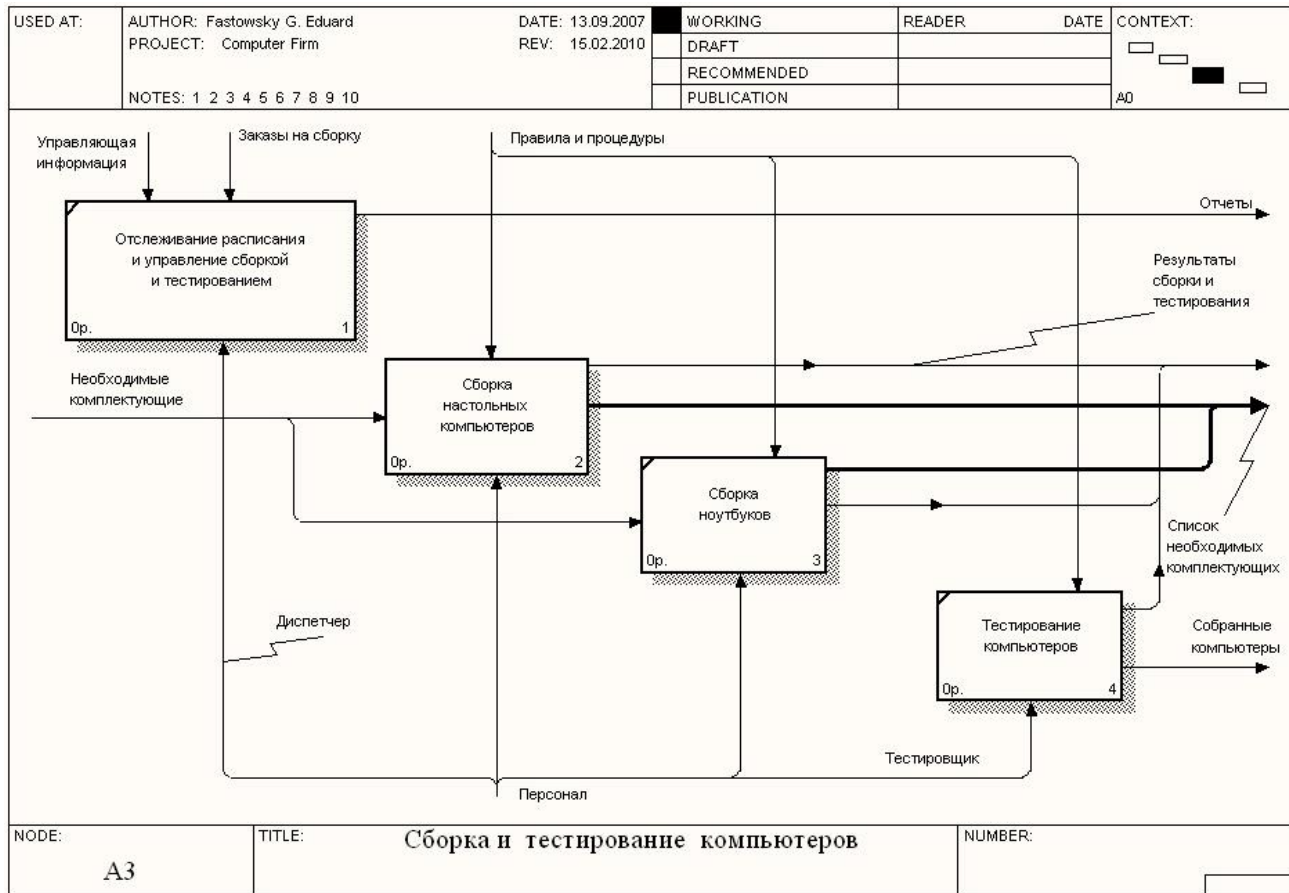


Рисунок 1

После соединения граничных стрелок с работами следующим шагом соединим работы между собой. Поступающие заказы на сборку сортируются диспетчером, после чего они поступают на вход управления работ "Сборка настольных компьютеров" и "Сборка ноутбуков" (стрелки *Заказы на настольные компьютеры* и *Заказы на ноутбуки*, соответственно). Когда компьютеры собраны, диспетчер дает указание на их отгрузку (стрелка *Указание передать компьютеры на отгрузку*).

Собранные компьютеры (выходы работ "Сборка настольных компьютеров" и "Сборка ноутбуков") должны быть протестированы, для чего они должны поступать на вход работы "Тестирование компьютеров" - стрелки *Настольные компьютеры* и *Ноутбуки*. После тестирования компьютеров отчет (стрелка *Результаты тестирования*) направляется диспетчеру, который дает указание отгрузить компьютеры.

Итоговая диаграмма декомпозиции показана на рисунке 2:

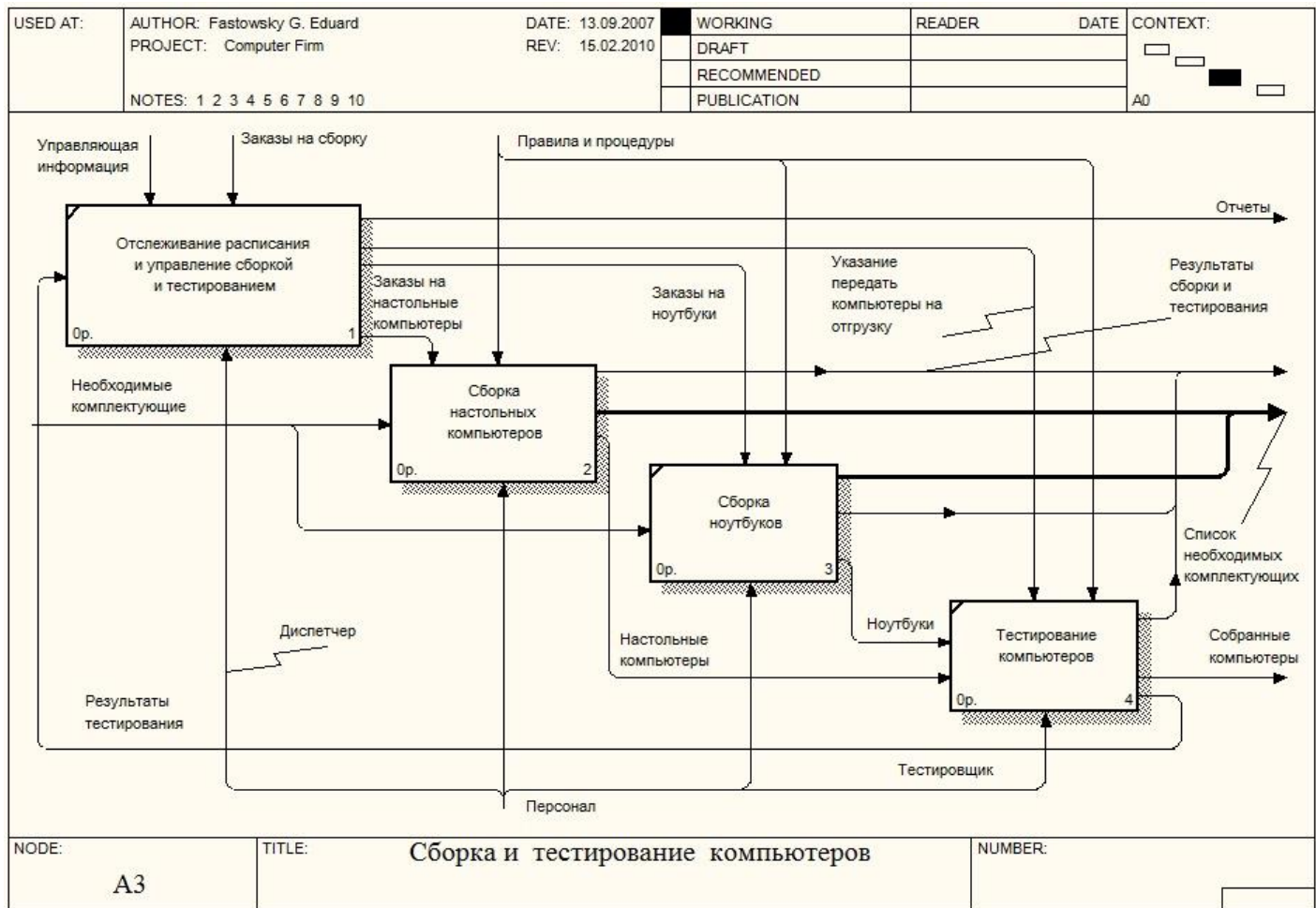


Рисунок 2

**Содержание отчета:**

- диаграмма декомпозиции
- краткое описание каждой работы

**Построение диаграммы декомпозиции в нотации IDEF3****Цель работы:**

- построить диаграмму декомпозиции в нотации IDEF3 одной из работ диаграмм IDEF0, построенных в предыдущих лабораторных работах

IDEF3 - методология моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. IDEF3 дает возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

Любая IDEF3-диаграмма может содержать работы, связи, перекрестки и объекты ссылок.

**Работа (Unit of Work, activity).** Изображается прямоугольником с прямыми углами (рис. 1) и имеет имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор); другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы (например, «Изготовление изделия»). Все стороны работы равнозначны. В каждую работу может входить и выходить ровно по одной стрелке.





Рисунок 1. Работа IDEF3


**Связи.** Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 возможны три вида связей:

| Изображение стрелки | Название                            | Описание   |
|---------------------|-------------------------------------|--|
|                     | Старшая (Precedence) стрелка        | сплошная линия, связывающая единицы работ (UOW). Рисуется слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется                           |
|                     | Потоки объектов (Object Flow)       | стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например когда объект порождается в одной работе и используется в другой |
|                     | Стрелка отношения (Relational Link) | пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок. Значение задается аналитиком отдельно для каждого случая          |

**Перекрестки (Junction).** Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления.

Типы перекрестков:

| Обозначение | Наименование                        | Смысл в случае слияния стрелок (Fan-in Junction)                   | Смысл в случае разветвления стрелок (Fan-out Junction)          |
|-------------|-------------------------------------|--|---|
|             | Асинхронное «И» (Asynchronous AND)  | Все предшествующие процессы должны быть завершены                  | Все следующие процессы должны быть запущены                     |
|             | Синхронное «И» (Synchronous AND)    | Все предшествующие процессы завершены одновременно                 | Все следующие процессы запускаются одновременно                 |
|             | Асинхронное «ИЛИ» (Asynchronous OR) | Один или несколько предшествующих процессов должны быть завершены  | Один или несколько следующих процессов должны быть запущены     |
|             | Синхронное «ИЛИ» (Synchronous OR)   | Один или несколько предшествующих процессов завершены одновременно | Один или несколько следующих процессов запускаются одновременно |

|   |  |  |  |
|---|--|--|--|
|  | Исключающее<br>«ИЛИ» XOR<br>(Exclusive OR) | Только один предшествующий<br>процесс завершен | Только один следующий<br>процесс запускается |
|---|--|--|--|

**Объект ссылки.** Объект ссылки в IDEF3 выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой. Они используются в модели для привлечения внимания читателя к каким-либо важным аспектам модели. При внесении объектов ссылок помимо имени следует указывать тип объекта ссылки (рис. 2).

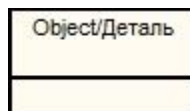


Рисунок 2. Объект ссылки

В данной лабораторной работе необходимо одну из работ, находящихся на диаграммах IDEF0, рассмотреть детально с помощью методологии IDEF3. При декомпозиции работы IDEF0 (и DFD) нужно учитывать, что стрелки на диаграммах IDEF0 или DFD означают потоки информации или объектов, передаваемых от одной работы к другой. На диаграммах IDEF3 стрелки могут показывать только последовательность выполнения работ, т.е. они имеют другой смысл, чем стрелки IDEF0 или DFD. Поэтому при декомпозиции работы IDEF0 или DFD в диаграмму IDEF3 стрелки не мигрируют на нижний уровень. Если необходимо показать на дочерней диаграмме IDEF3 те же объекты, что и на родительских диаграммах IDEF0 или DFD, необходимо использовать объекты ссылки.

Проведем декомпозицию работы *Сборка настольных компьютеров* диаграммы А3 "Сборка и тестирование компьютеров". Данная работа начинает выполняться, когда поступают заказы на сборку. Первым действием проверяется наличие необходимых для сборки комплектующих и заказ со склада отсутствующих. Далее комплектующие подготавливаются для последующей сборки (освобождение от упаковки, снятие заглушек и т.п.). Следующим шагом начинается непосредственно сам процесс сборки: установка материнской платы в корпус и процессора на материнскую плату, установка ОЗУ и винчестера. Данные действия выполняются всегда, независимо от конфигурации компьютера. Далее по желанию клиента могут быть установлены некоторые дополнительные комплектующие - DVD привод, ТВ-тюнер, кард-ридер. На этом сборка компьютера завершается. Следующим шагом идет установка операционной системы. По желанию клиента также может быть установлено дополнительное программное обеспечение. Последним действием составляется отчет о проделанной работе.

Выделим работу *Сборка настольных компьютеров* диаграммы А3 "Сборка и тестирование компьютеров", нажмем на кнопку "Go to Child Diagram" панели инструментов и выберем нотацию IDEF3. Дочерние работы всегда можно добавить на диаграмму в процессе ее построения, поэтому число дочерних работ оставим по умолчанию. При создании дочерней диаграммы BPWin переносит граничные стрелки родительской работы, их необходимо удалить и заменить на объекты ссылок. Заменяем стрелки "Заказы на настольные компьютеры", "Необходимые комплектующие", "Список необходимых комплектующих", "Настольные компьютеры" и "Результаты сборки" на объекты ссылок - кнопка "Referent" на панели инструментов, в появившемся окне выбрать переключатель "Arrow" и выбрать из списка нужное название (рис. 3):

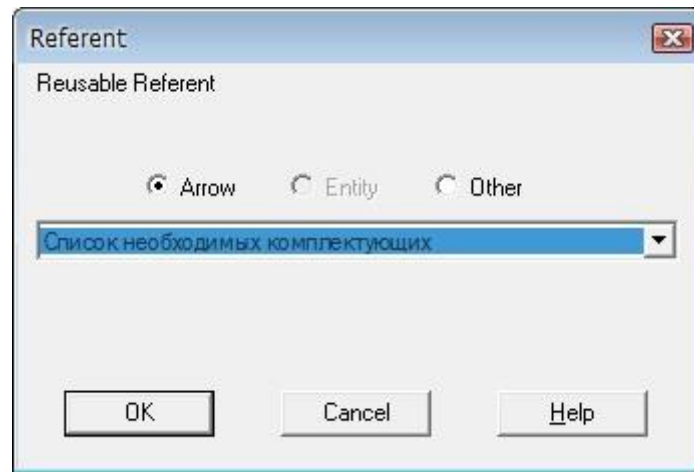


Рисунок 3. Добавление объекта ссылки

Далее начинаем располагать на диаграмме работы, отражающие указанные выше действия, выполняемые при сборке компьютеров. Итоговая диаграмма декомпозиции работы в нотации IDEF3 имеет вид:

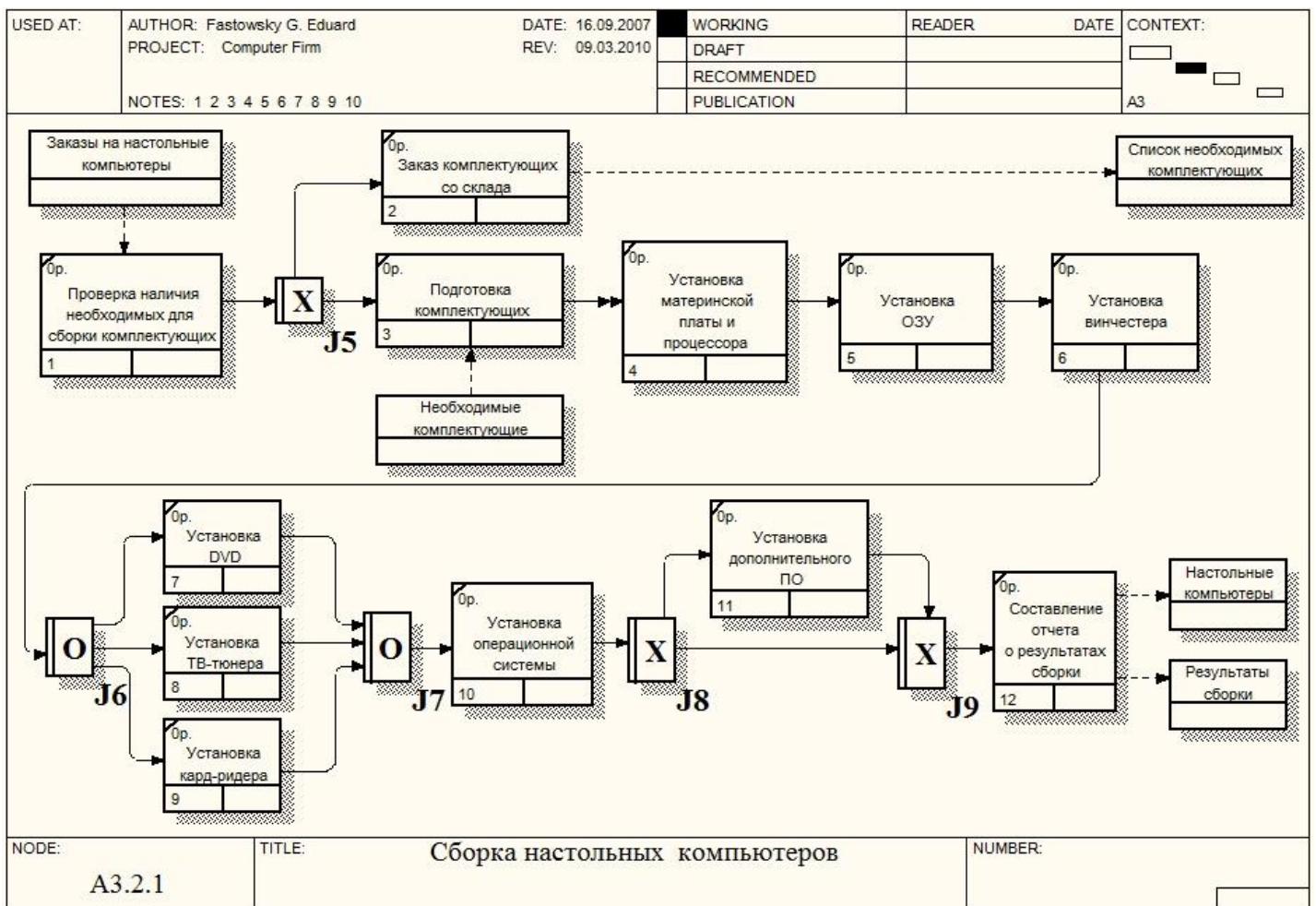


Рисунок 4. Диаграмма декомпозиции

Рассмотрим основные особенности этой диаграммы. После проверки наличия необходимых для сборки комплектующих возможно одно из двух действий - или заказ со склада недостающих комплектующих, или, если все комплектующие в наличии, их подготовка. Поэтому мы поставили перекресток разветвления типа "Исключающее ИЛИ". Работы "Подготовка комплектующих" и "Установка материнской платы и процессора" соединены связью "Поток объектов". Тем самым мы показываем, что между этими работами передаются объекты. Все последующие работы соединяются связями "старшая стрелка", поскольку они только показывают последовательность действий над одними и теми же объектами.

После установки винчестера возможна установка DVD привода, ТВ-тюнера, кардридера или любая их комбинация. Поэтому мы поставили перекресток разветвления типа "Асинхронное ИЛИ". Такой же перекресток стоит и после завершения этих работ. Далее после установки операционной системы может быть установлено дополнительное ПО, или же сразу формируется отчет, поэтому мы поставили перекресток разветвления типа "Исключающее ИЛИ". За перекрестком разветвления типа "Исключающее ИЛИ" может следовать только такой же перекресток слияния, поэтому перед работой "*Составление отчета о результатах сборки*" мы поставили такой же.

#### Содержание отчета:

- краткое описание декомпозируемой работы
- диаграмма декомпозиции

#### Построение диаграммы декомпозиции в нотации DFD

#### Цель работы:

- построить диаграмму декомпозиции в нотации DFD одной из работ диаграмм IDEF0, построенных в предыдущих лабораторных работах

Диаграммы потоков данных (Data flow diagram, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет моделируемую систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. Главная цель DFD - показать, как каждая работа преобразует свои входные данные в выходные, а также выявить отношения между этими работами.

Любая DFD-диаграмма может содержать работы, внешние сущности, стрелки (потоки данных) и хранилища данных.

**Работы.** Работы изображаются прямоугольниками с закругленными углами (рис. 1), смысл их совпадает со смыслом работ IDEF0 и IDEF3. Так же как работы IDEF3, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0. Все стороны работы равнозначны. В каждую работу может входить и выходить по несколько стрелок.



Рисунок 1. Работа в DFD

**Внешние сущности.** Внешние сущности изображают входы в систему и/или выходы из нее. Одна внешняя сущность может одновременно предоставлять входы (функционируя как поставщик) и принимать выходы (функционируя как получатель). Внешняя сущность представляет собой материальный объект, например заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что они находятся за пределами границ анализируемой системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы (рис. 2).



Рисунок 2. Внешняя сущность в DFD



**Стрелки (потoki данных).** Стрелки описывают движение объектов из одной части системы в другую (отсюда следует, что диаграмма DFD не может иметь граничных стрелок). Поскольку все стороны работы в DFD равнозначны, стрелки могут начинаться и заканчиваться на любой стороне прямоугольника. Стрелки могут быть двунаправлены.

**Хранилище данных.** В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое (рис. 3). Хранилище данных - это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми. Оно в общем случае является прообразом будущей базы данных, и описание хранящихся в нем данных должно соответствовать информационной модели (Entity-Relationship Diagram).



Рисунок 3. Хранилище данных в DFD

**Декомпозиция работы IDEF0 в диаграмму DFD.** При декомпозиции работы IDEF0 в DFD необходимо выполнить следующие действия:

- удалить все граничные стрелки на диаграмме DFD;
- создать соответствующие внешние сущности и хранилища данных;
- создать внутренние стрелки, начинающиеся с внешних сущностей вместо граничных стрелок;
- стрелки на диаграмме IDEF0 затоннелировать

Строго придерживаться правил нотации DFD не всегда удобно, поэтому BPWin позволяет создавать в DFD диаграммах граничные стрелки.

**Построение диаграммы декомпозиции.** Проведем декомпозицию работы *Отгрузка и снабжение* диаграммы A0 "Деятельность предприятия по сборке и продаже компьютеров и ноутбуков". В этой работе мы выделили следующие дочерние работы:

- снабжение необходимыми комплектующими - занимается действиями, связанными с поиском подходящих поставщиков и заказом у них необходимых комплектующих
- хранение комплектующих и собранных компьютеров
- отгрузка готовой продукции - все действия, связанные с упаковкой, оформлением документации и собственно отгрузкой готовой продукции

Выделим работу *Отгрузка и снабжение* диаграммы A0 "Деятельность предприятия по сборке и продаже компьютеров и ноутбуков", нажмем на кнопку "Go to Child Diagram" панели инструментов и выберем нотацию DFD. При создании дочерней диаграммы BPWin переносит граничные стрелки родительской работы, их необходимо удалить и заменить на внешние сущности. Стрелки механизмов, стрелки управления "Правила и процедуры", "Управляющая информация" и стрелку выхода "Отчеты" на дочерней диаграмме задействованы не будут, чтоб не загромождать диаграмму менее существенными деталями. Остальные стрелки заменим на внешние сущности - кнопка "External Reference Tool" на панели инструментов, в появившемся окне выбрать переключатель "Arrow" и выбрать из списка нужное название (рис. 4):

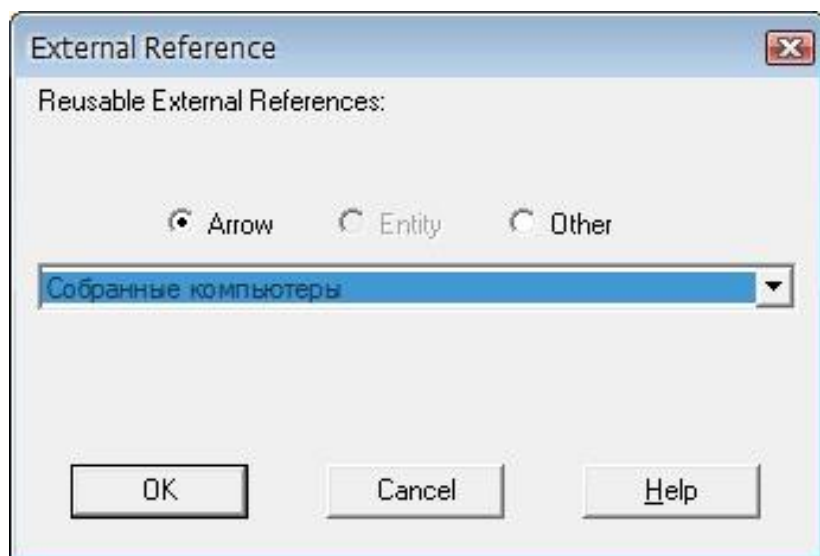


Рисунок 4. Добавление внешней сущности

Далее разместим дочерние работы, свяжем их со внешними сущностями и между собой (рис. 5):

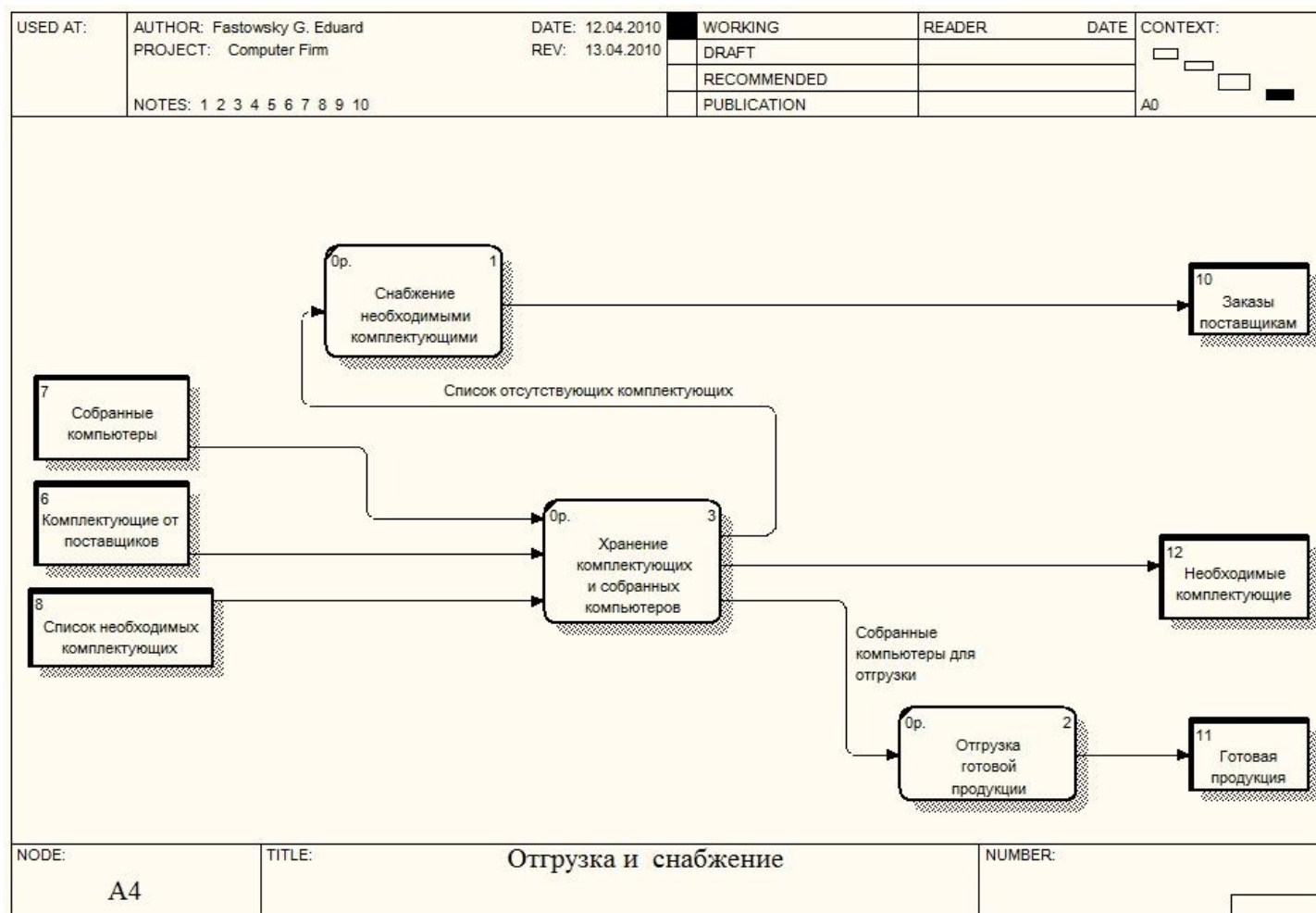


Рисунок 5. Работы и внешние сущности

Центральной здесь является работа "Хранение комплектующих и собранных компьютеров". На ее вход поступают собранные компьютеры и полученные от поставщиков комплектующие, а также список необходимых для сборки компьютеров комплектующих. Выходом этой работы будут необходимые комплектующие (если они есть в наличии), список отсутствующих комплектующих, передаваемый на вход работы "Снабжение необходимыми комплектующими" и собранные компьютеры, передаваемые на отгрузку. Выходами работ "Снабжение

необходимыми комплектующими" и

"Отгрузка готовой продукции" будут,

соответственно, заказы поставщикам и готовая продукция.

Следующим шагом необходимо определить, какая информация необходима для каждой работы, т.е. необходимо разместить на диаграмме хранилища данных (рис. 6).

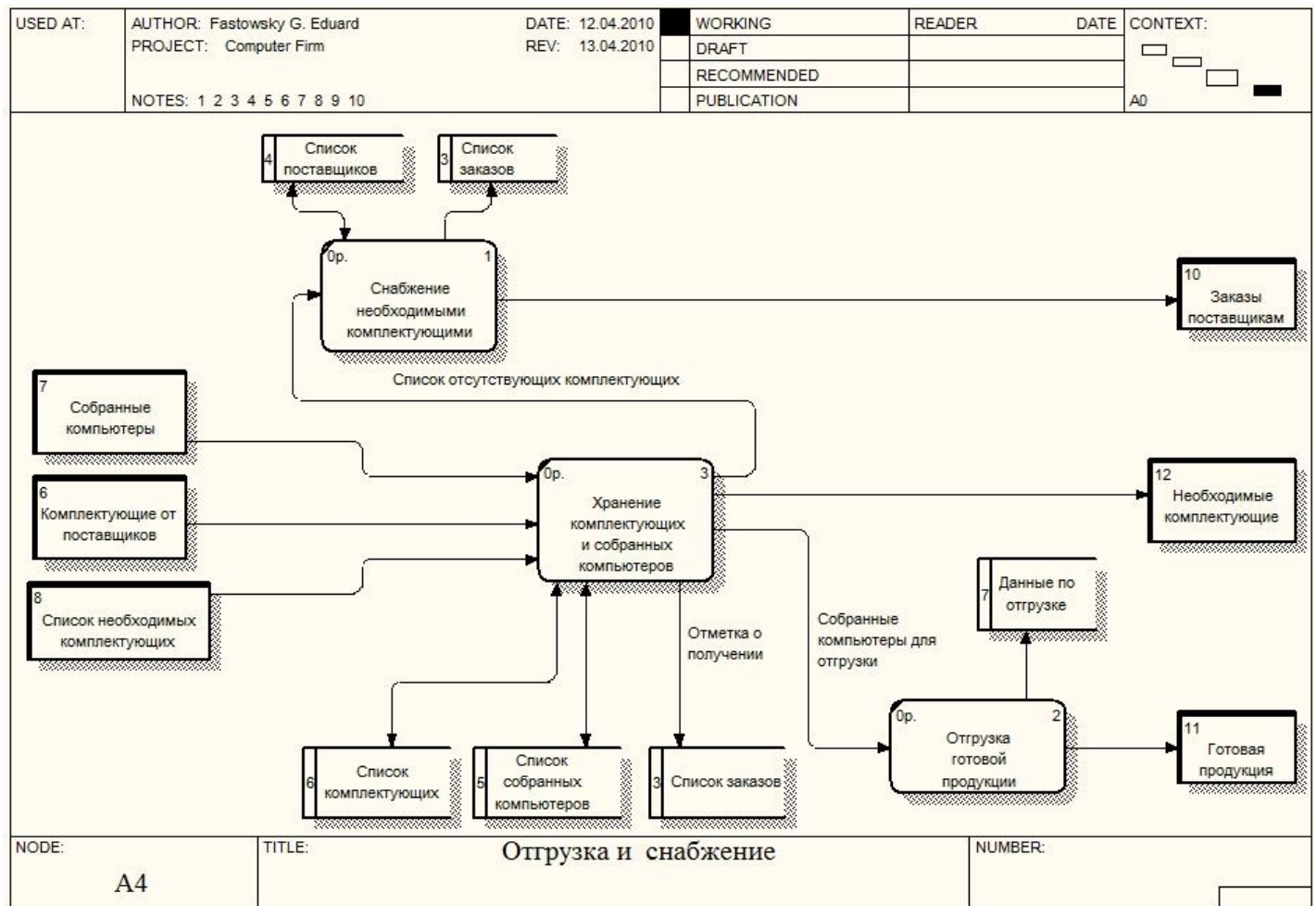


Рисунок 6. Итоговая диаграмма декомпозиции

Работа "Снабжение необходимыми комплектующими" работает с информацией о поставщиках и с информацией о заказах, сделанных у этих поставщиков. Стрелка, соединяющая работу и хранилище данных "Список поставщиков" двунаправленная, т.к. работа может как получать информацию о имеющихся поставщиках, так и вносить данные о новых поставщиках. Стрелка, соединяющая работу с хранилищем данных "Список заказов" однонаправленная, т.к. работа только вносит информацию о сделанных заказах.

Работа "Хранение комплектующих и собранных компьютеров" работает с информацией о получаемых и выдаваемых комплектующих и собранных компьютерах, поэтому стрелки, соединяющая работу с хранилищами данных "Список комплектующих" и "Список собранных компьютеров" двунаправленные. Также эта работа при получении комплектующих должна делать отметку о том, что заказ поставщикам выполнен. Для этого она связана с хранилищем данных "Список заказов" однонаправленной стрелкой. Обратите внимание, что на DFD диаграммах одно и тоже хранилище данных может дублироваться.

Наконец, работа "Отгрузка готовой продукции" должна хранить информацию по выполненным отгрузкам. Для этого вводится соответствующее хранилище данных - "Данные по отгрузке". Последним действием необходимо стрелки родительской работы затуннелировать (рис. 7):

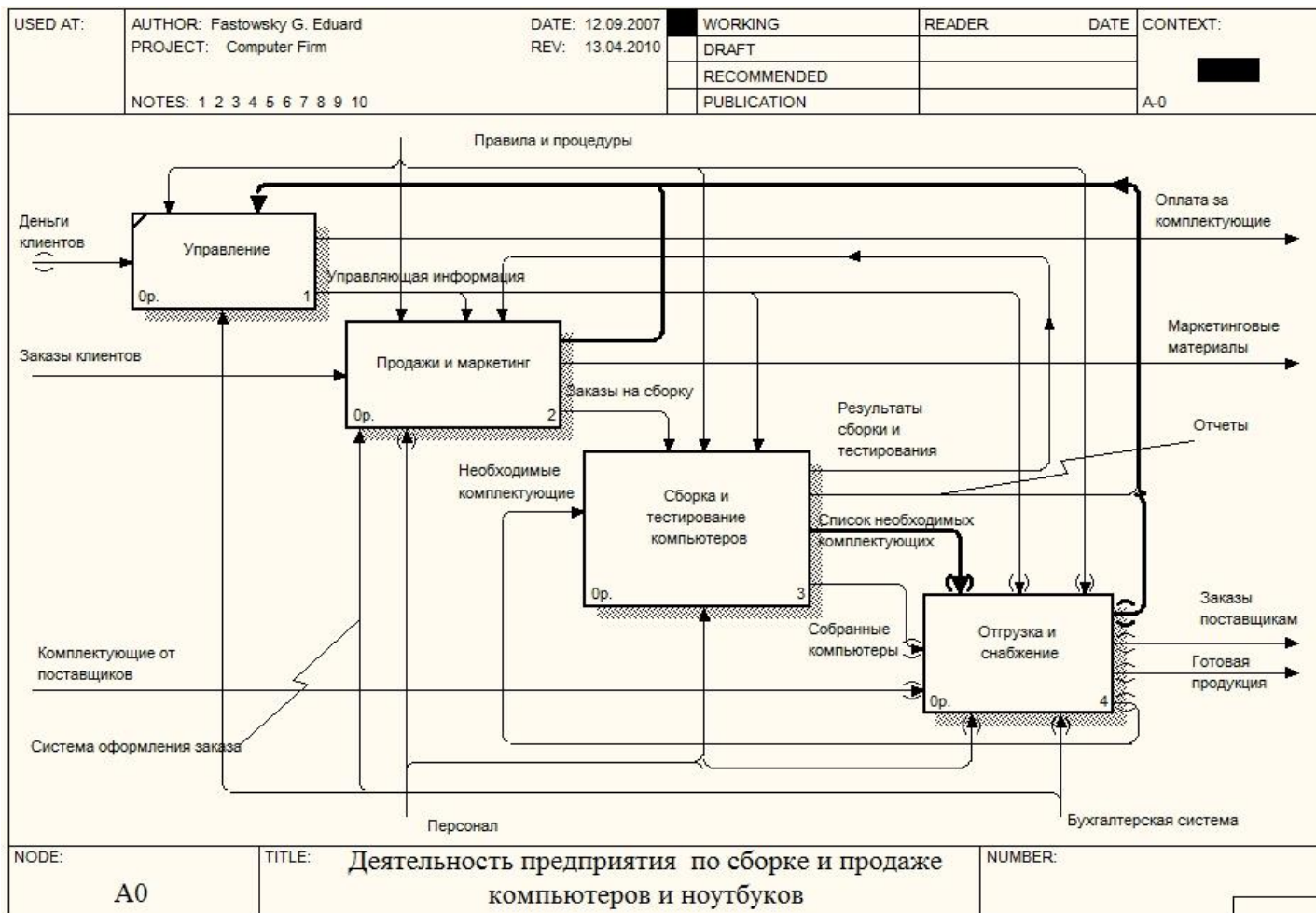


Рисунок 7. Диаграмма IDEF0 с затуннелированными стрелками работы "Отгрузка и снабжение"  
**Содержание отчета:**

- краткое описание декомпозируемой работы  
 диаграмма декомпозиции



## AllFusion Process Modeler

## Построение FEO диаграмм и диаграмм дерева узлов

**Цель работы:**

- построить FEO диаграмму для одной из имеющихся диаграмм
- построить диаграмму дерева узлов
- 

## FEO диаграммы

FEO (For Exposition Only) диаграммы (другое название - диаграммы только для экспозиции, описания) используются для иллюстрации альтернативной точки зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0. FEO диаграммы позволяют нарушить любое синтаксическое правило, поскольку эти диаграммы - фактически обычные картинки - копии стандартных диаграмм. Например, работа на FEO диаграмме может не иметь стрелок выхода или управления. AllFusion Process Modeler позволяет также строить FEO диаграммы для диаграмм в нотации DFD.

Для построения FEO диаграммы необходимо выбрать пункт меню *Diagram -> Add FEO Diagram* и в появившемся окне выбрать диаграмму, на базе которой будет строиться FEO диаграмма (рис. 1).

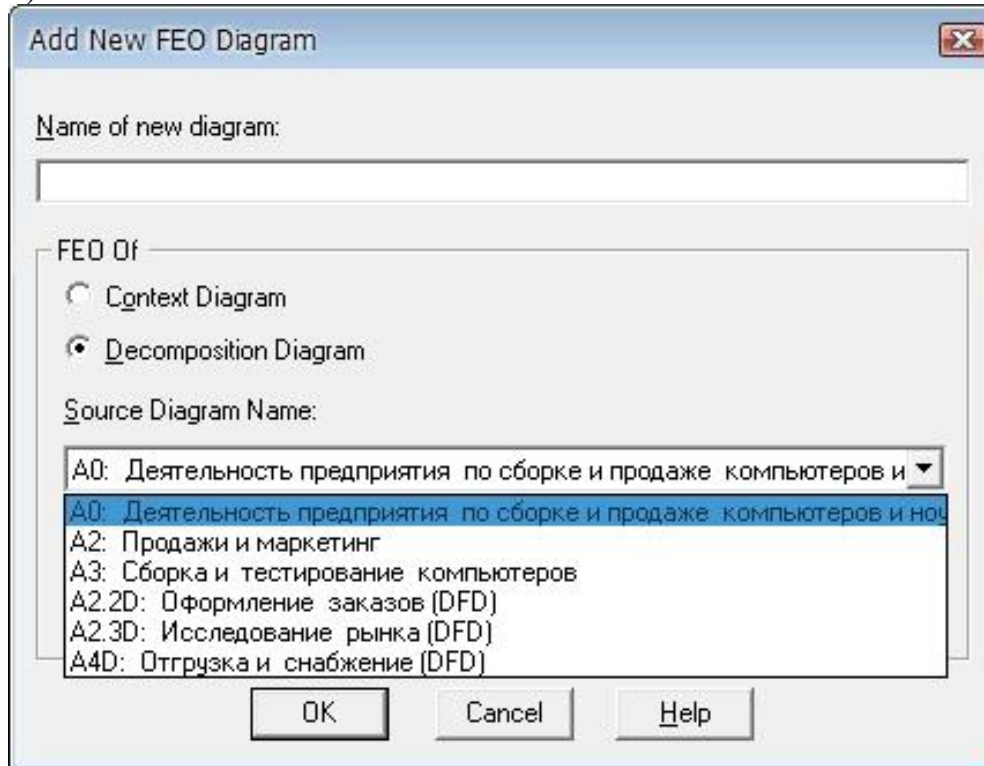


Рисунок 1. Добавление FEO диаграммы

Созданная диаграмма будет точной копией родительской диаграммы и будет иметь номер, равный номеру родительской диаграммы + буква F. После создания диаграммы ее можно изменять. При этом изменения не будут влиять на родительскую диаграмму.

Для просмотра списка имеющихся FEO диаграмм нужно выбрать в *Обозревателе Модели (Model Explorer)* вкладку *Diagrams* (рис.2).

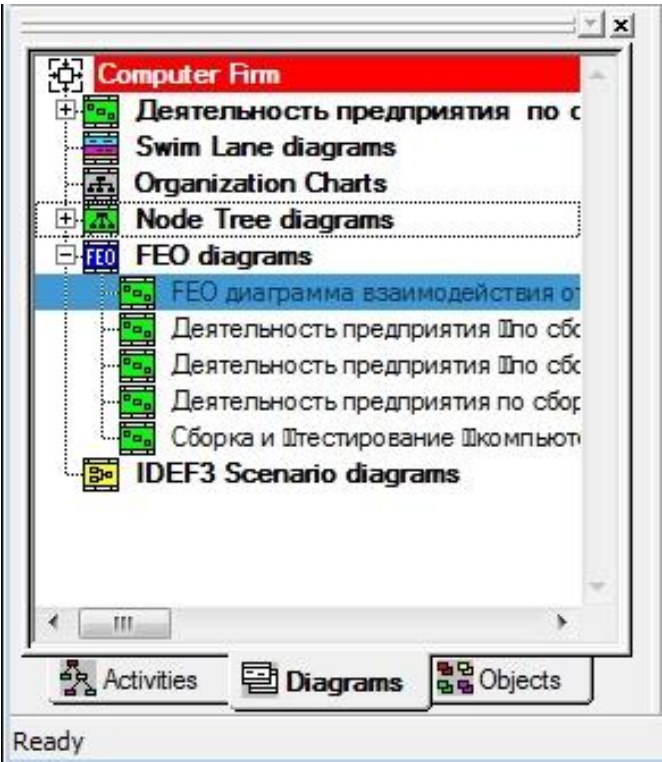


Рисунок 2. Просмотр списка имеющихся FEO диаграмм

Построим FEO диаграмму для диаграммы декомпозиции второго уровня A0 "Деятельность предприятия по сборке и продаже компьютеров и ноутбуков" и покажем на ней как дочерние работы связаны между собой. Для этого создаем диаграмму, как показано выше, и удаляем на ней все граничные стрелки. Итоговая FEO диаграмма показана на рис.3:

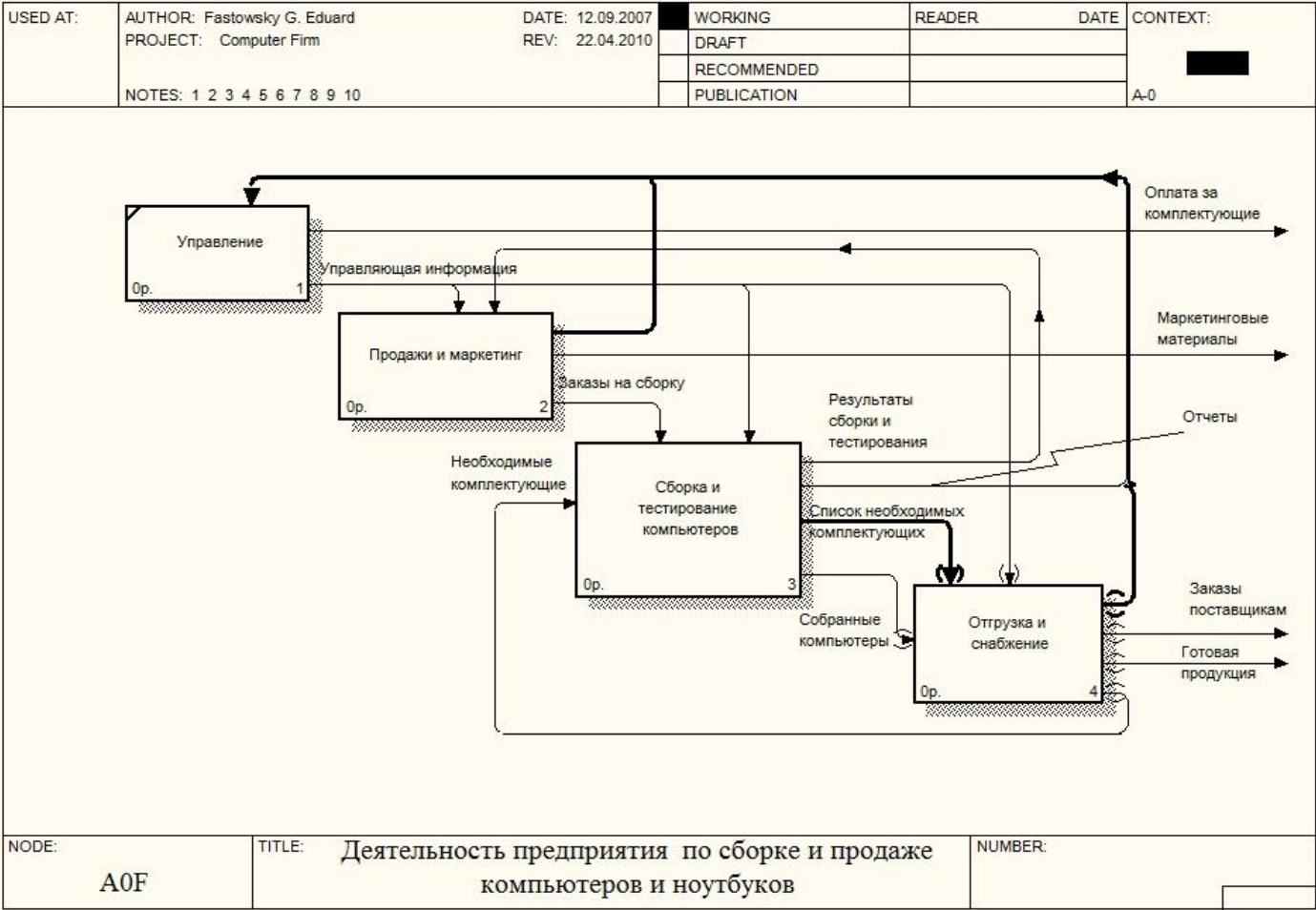


Рисунок 3. FEO диаграмма

## Диаграммы дерева узлов

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. В одной модели диаграмм дерева узлов может быть множество, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

Для построения диаграммы дерева узлов необходимо выбрать пункт меню *Diagram -> Add Node Tree*. Появляется мастер, с помощью которого диаграмма будет создана. На первом шаге (рис.4) задается имя диаграммы дерева узлов, узел верхнего уровня и глубина дерева. Имя дерева узлов по умолчанию совпадает с именем работы верхнего уровня, а номер диаграммы генерируется автоматически как номер узла верхнего уровня + буква N.

Рисунок 4. Создание диаграммы дерева узлов. Шаг 1

На втором шаге мастера (рис.5) задаются свойства диаграммы дерева узлов.

Рисунок 5. Создание диаграммы дерева узлов. Шаг 2

По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные работы - в виде прямоугольников. Если необходимо отобразить все дерево в виде прямоугольников, то следует снять галочку возле опции "Bullet last level". Список всех созданных диаграмм дерева

узлов можно посмотреть в *Обозреватели*

*Модели.*

Диаграмма дерева узлов для всех узлов модели показана на рис. 6:

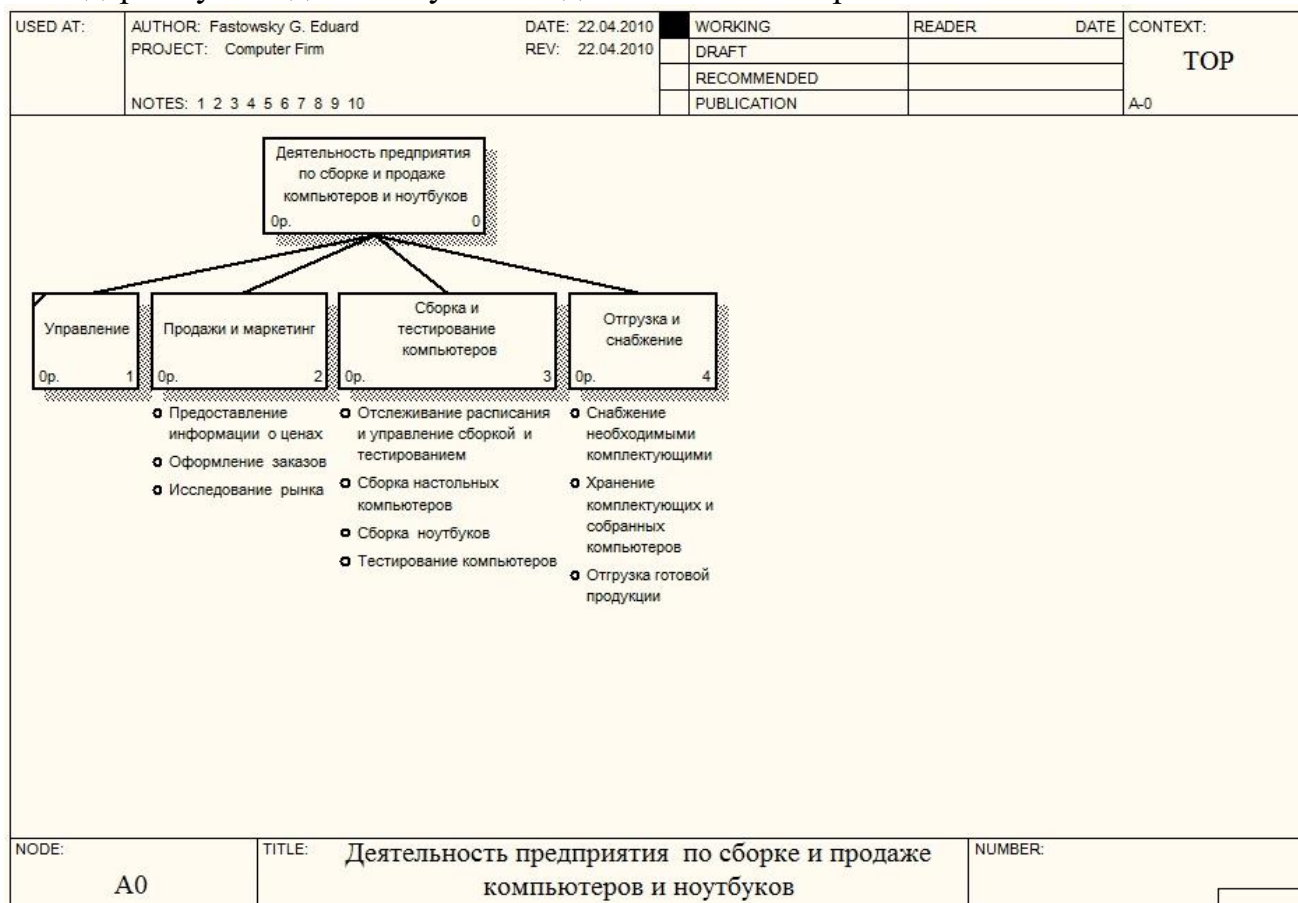


Рисунок 6. Диаграммы дерева узлов

### Содержание отчета:

- ФЕО диаграмма
- диаграмма дерева узлов

## 12. Лабораторная работа 6. Использование системы контроля версий git совместно с сайтом github

### Задание

**Ознакомиться с теоретическим материалом.**

**Зарегистрировать личный аккаунт на сайте github.com**

**С помощью клиента git создать соединение с github и создать проект, содержащий не менее 2 ветвей.**

**Оформить отчет со скриншотами, подтверждающими выполненные работы.**

**Отчёт представить в электронном виде через сайт moodle.iubip.ru.**

### Теоретический материал.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome).

Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности (включая SSL), а для частных проектов предлагаются различные платные тарифные планы.

Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». На футболках же печатают совсем другую фразу: «Fork you!» («Ответвись!»). С одной стороны, она созвучна с англоязычным ругательством и намекает на неформальную атмосферу. С другой, эти слова напоминают, что создавать новые форки с Git можно легко и безболезненно — традиционно, к созданию веток разработчики проектов с открытым исходным кодом относятся негативно — а также созвучна названию одной из возможностей Github — очереди форков.

Талисманом GitHub выбран осьмикот (англ. octocat), который, вопреки распространённому заблуждению, не имеет отношения к короткометражке «Octocat Adventure», а просто был найден Томом Престон-Вернером на iStock Photo.

#### Возможности

Создатели сайта называют GitHub «социальной сетью для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева. (Пошаговая инструкция по работе git и github для студентов)

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

- На платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей.
- Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.
- Код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта.
- Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.
- На сайте есть pastebin-сервис gist.github.com для быстрой публикации фрагментов кода.

Ранее Ruby-проекты могли быть автоматически опубликованы в RubyGems-репозитории сервиса, но в октябре 2009 GitHub отказался от этого сервиса.

#### Популярность

Первый частный репозиторий был создан 12 января 2008. К концу 2011 года в проекте уже было зарегистрировано более миллиона пользователей и более двух миллионов репозиторий. По состоянию на март 2017 года на сайте существовало более 58 миллионов репозиторий.

В конце 2008 года GitHub получил награду как «Лучший стартап-дебют».

GitHub наиболее популярен среди Ruby-разработчиков[нет в источнике]. Кроме того, многие крупные IT-компании размещают свои официальные репозитории на этом сервисе:

- Facebook
- Twitter
- Google
- HP webOS
- Yahoo
- Perl
- Erlang
- Scala
- Ruby on Rails
- CyanogenMod
- PHP
- JUnit
- jQuery
- Prototype
- MooTools
- Microsoft
- osCommerce
- Valve
- Apple

На GitHub размещают свои открытые проекты федеральные агентства США.

Сайт github.com позиционируется как веб-сервис хостинга проектов с использованием системы контроля версий git, а также как социальная сеть для разработчиков. Пользователи могут создавать неограниченное число репозиторий, для каждого из которых предоставляется wiki, система issue tracking-a, есть возможность проводить code review и многое другое. GitHub на данный момент **является** самым популярным сервисом такого рода, обогнав Sourceforge и Google Code.

Для open-source проектов использование сайта бесплатно. При необходимости иметь приватные репозитории, есть возможность перейти на платный тарифный план:

**Free for open source**  
Unlimited public repositories and unlimited public collaborators  
[Create a free account](#)

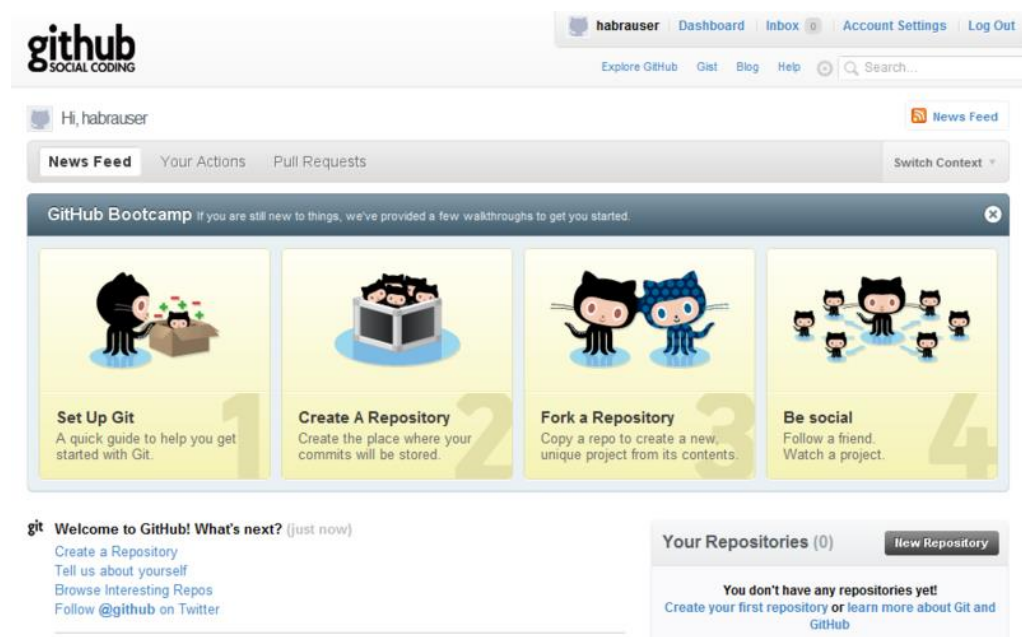
| Plan   | Price   | Private Repositories | Private Collaborators | Public Repositories | Public Collaborators | Action                            |
|--------|---------|----------------------|-----------------------|---------------------|----------------------|-----------------------------------|
| Micro  | \$7/mo  | 5                    | 1                     | Unlimited           | Unlimited            | <a href="#">Create an account</a> |
| Small  | \$12/mo | 10                   | 5                     | Unlimited           | Unlimited            | <a href="#">Create an account</a> |
| Medium | \$22/mo | 20                   | 10                    | Unlimited           | Unlimited            | <a href="#">Create an account</a> |

**Business Plans**

| Plan     | Price    | Private Repositories | Teams     | Public Repositories | Action                                 |
|----------|----------|----------------------|-----------|---------------------|--|
| Bronze   | \$25/mo  | 10                   | Unlimited | Unlimited           | <a href="#">Create an organization</a> |
| Silver   | \$50/mo  | 20                   | Unlimited | Unlimited           | <a href="#">Create an organization</a> |
| Gold     | \$100/mo | 50                   | Unlimited | Unlimited           | <a href="#">Create an organization</a> |
| Platinum | \$200/mo | 125                  | Unlimited | Unlimited           | <a href="#">Create an organization</a> |



Начнем с регистрации. Идем по ссылке [github.com/signup/free](https://github.com/signup/free) и вводим свои данные. После регистрации мы попадаем на Dashboard нашего аккаунта:



Сейчас у нас нет ни одного репозитория, и мы можем либо создать новый репозиторий, либо ответвиться (fork) от уже существующего чужого репозитория и вести собственную ветку разработки. Затем, при желании, свои изменения можно предложить автору исходного репозитория (Pull request).

Но для начала установим git и настроим его для работы с сайтом.

### Установка системы контроля версий Git

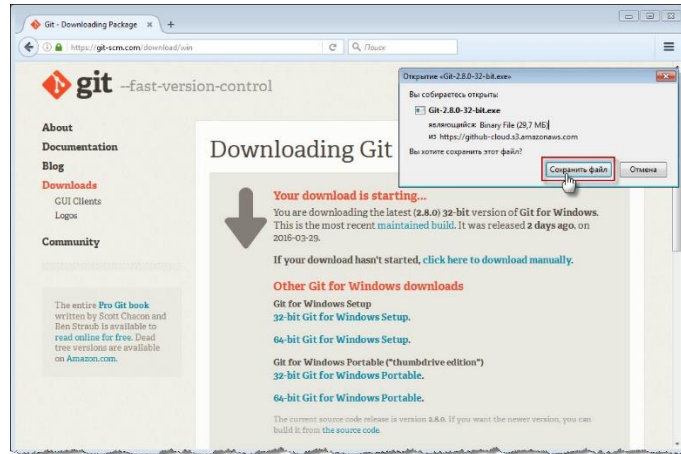
**Git** – свободно распространяемая мощная и быстрая распределённая система контроля версий с открытым кодом, предназначенная для работы с любыми, как очень маленькими, так и очень большими проектами.

В данном документе описан процесс установки **Git** на ваш компьютер с операционной системой Windows. Описание установки приведено на примере релиза 2.8.0.

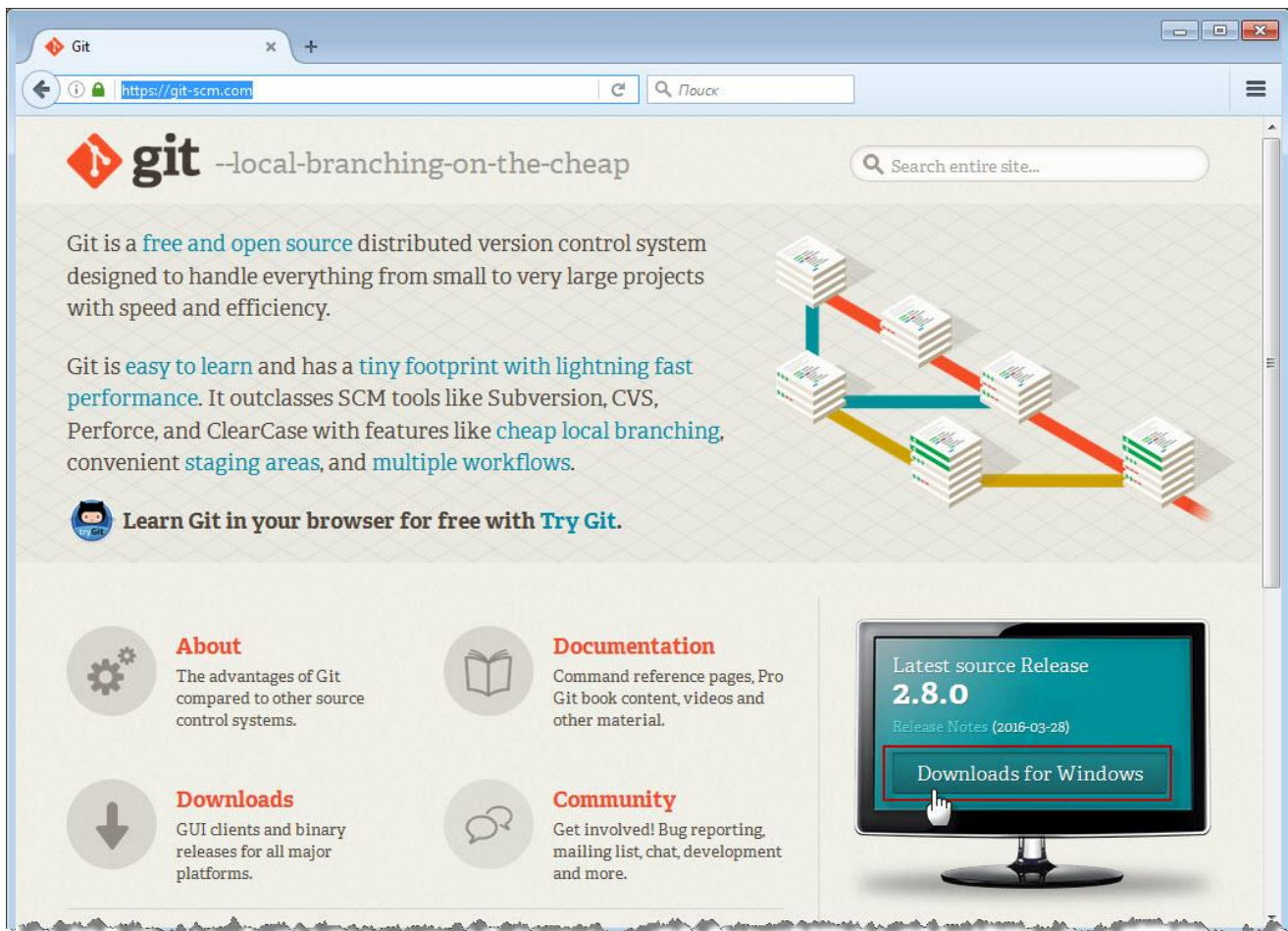
### Загрузка файла установки

Для загрузки файла установки вам необходимо открыть в веб-браузере страницу <https://git-scm.com/> и нажать на кнопку .

Откроется окно загрузки файла установки

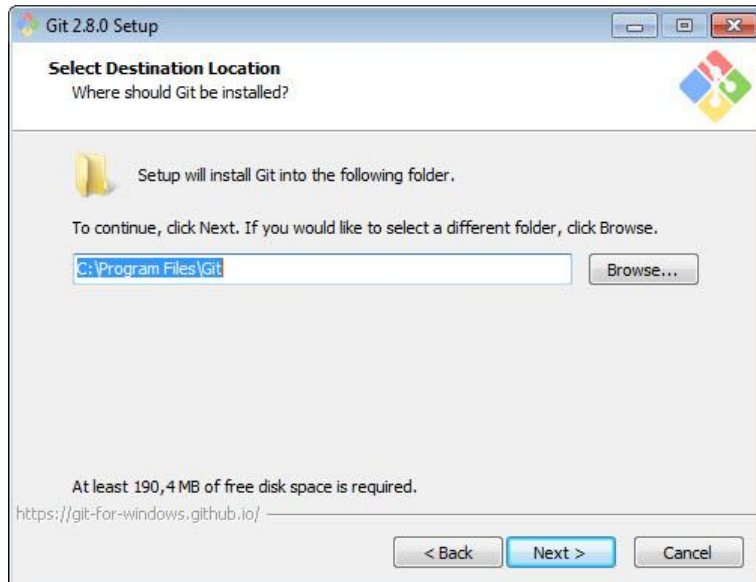


Программа установки автоматически определяет тип операционной системы и предлагает сохранить соответствующий файл установки. Однако вы можете отказаться от скачивания предлагаемого файла, нажав на кнопку **Отмена**, и выбрать другой файл для установки.



Если предлагаемый для загрузки файл вас устраивает, нажмите на кнопку **Сохранить файл**.  
Файл будет помещён в папку, назначенную для загрузки файлов.

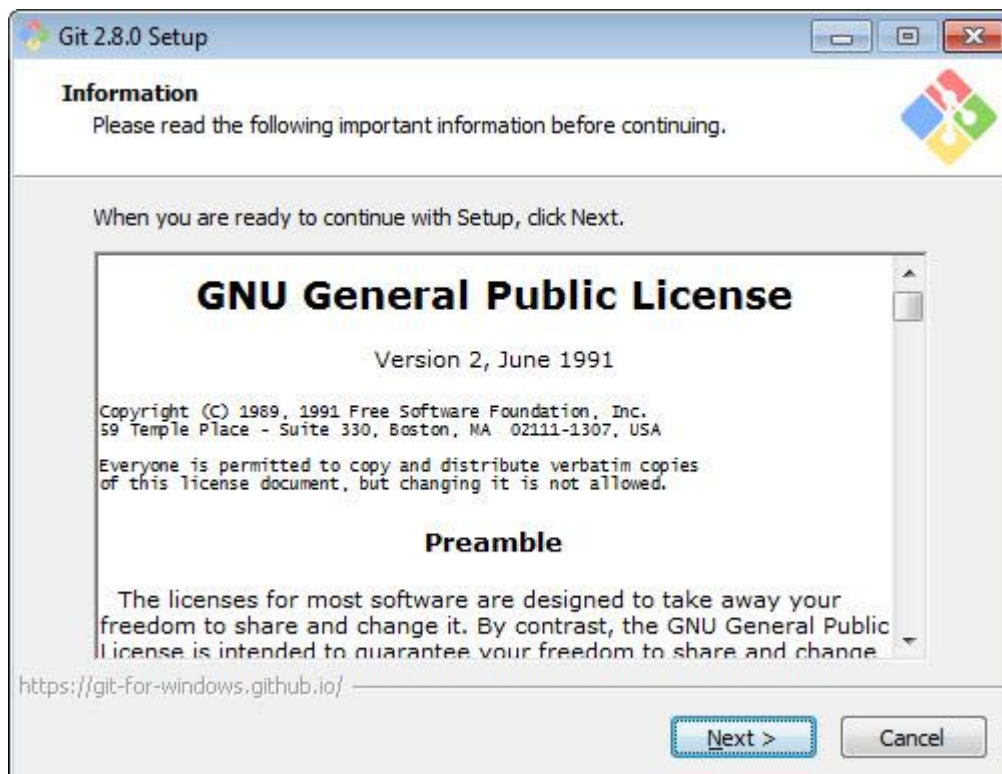





## Установка Git

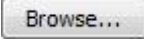
После того, как файл загрузится, можно начать установку. Для этого:

Запустите на выполнение загруженный файл (в нашем примере файл Git-2.8.0-32-bit.exe). Откроется окно с текстом лицензионного соглашения. **Git** распространяется в рамках *GNU General Public License* - универсальной общественной лицензии GNU, по


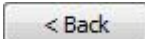
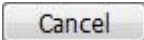


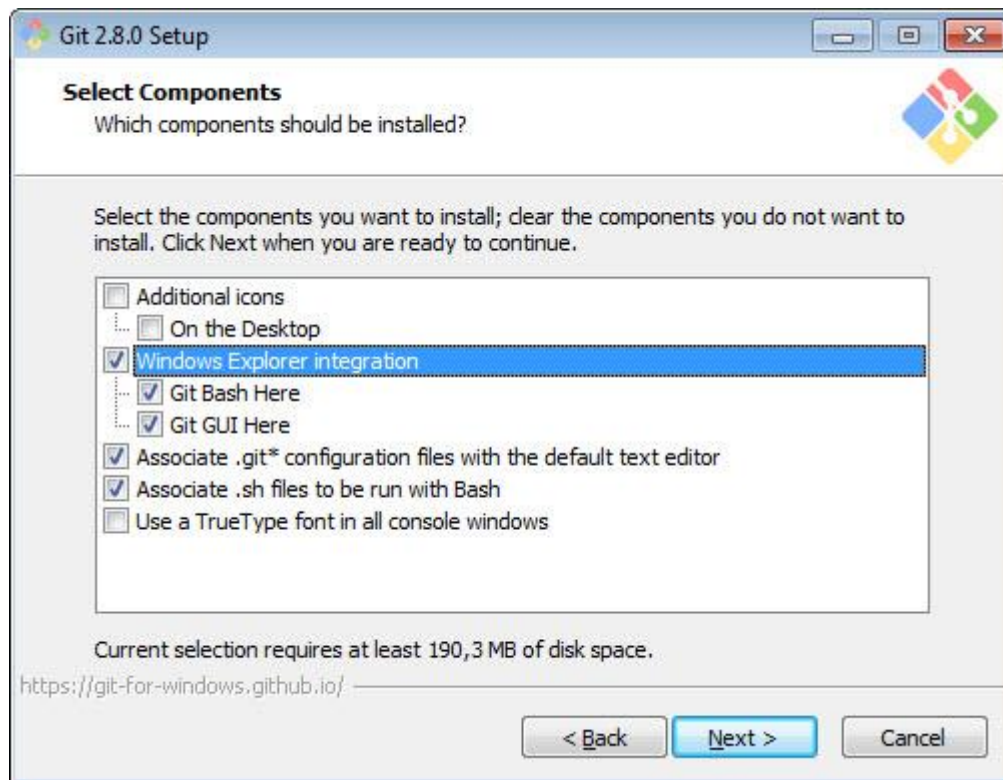
которой программное обеспечение передаётся автором в общественную собственность.

Прочтите текст лицензионного соглашения, после чего щёлкните по кнопке . Откроется окно выбора папки установки.

Вы можете выбрать любую другую папку, имеющуюся на диске, с помощью стандартной процедуры Windows, щёлкнув по кнопке  или ввести путь в строке «C:\Program Files\Git».

Рекомендуется оставить папку, предложенную по умолчанию.

В нижней части окна приведены требования к объёму свободного места на диске. Для продолжения установки щёлкните по кнопке . Практически на любом этапе установки вы можете вернуться к предыдущему шагу, щёлкнув по кнопке  или прекратить процесс установки, щёлкнув по кнопке .



Откроется окно для выбора устанавливаемых компонентов.

Установкой/снятием флагов сформируйте для себя набор компонентов для установки:

- **Additional icons** - вывод на рабочий стол значков для запуска установленных компонентов;
- **Windows Explore Integration** - установка способов вызова **Git** командой контекстного меню на папке (правой кнопкой мыши):

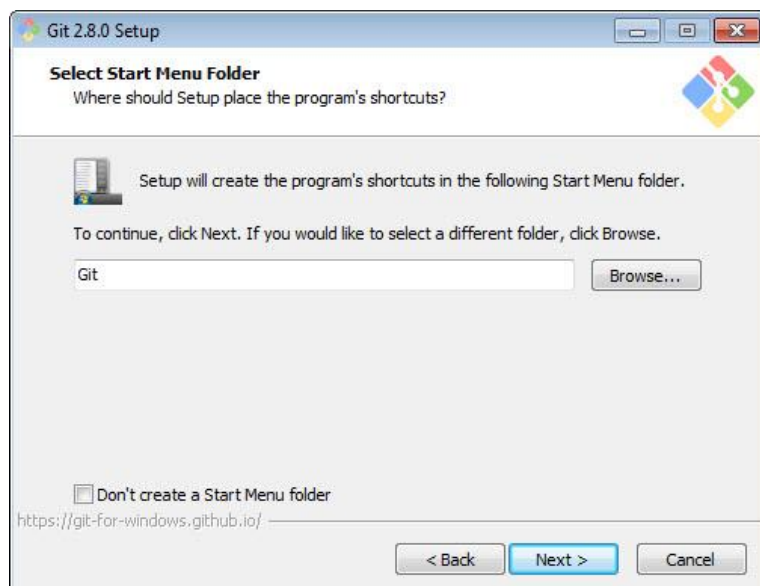
• **Git Bash Here** - запуск **Git** в виде консоли из командной строки;

• **Git GUI Here** - запуск **Git** в виде графического приложения.

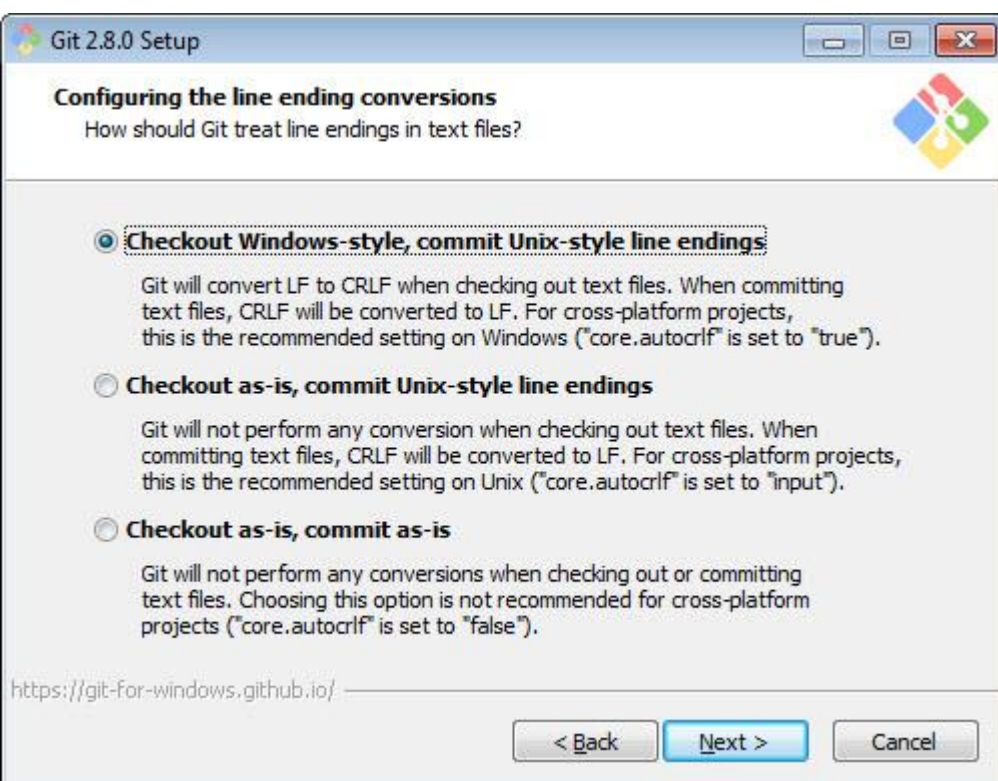
○ **Associate .git\* configuration files with the default text editor** - при установленном флаге файлы `.git*` будут автоматически открываться для редактирования текстовым редактором, установленным по умолчанию;

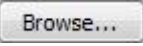
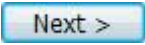
○ **Associate .sh files to be run with Bash** - при установленном флаге файлы `.sh*` будут автоматически открываться в командной оболочке *Bash*;

○ **Use a TrueType font in all console windows** - при установленном флаге во всех консольных окнах будет использоваться шрифт TrueType. По окончании конфигурирования набора компонентов для продолжения установки щёлкните по кнопке **Next >**.



Откроется окно выбора папки для размещения ярлыков запуска программ.



По умолчанию предлагается папка «Git» («C:\Program Files\Git»). Вы можете выбрать любую другую папку, щёлкнув по кнопке . Можно вовсе отказаться от создания ярлыков запуска, установив флаг «**Don't create a Start Menu Folder**». Для продолжения установки щёлкните по кнопке .

Откроется окно настройки переменных окружения.



Предлагаются три варианта запуска **Git**. В зависимости от выбранного способа программа установки внесёт соответствующие изменения в переменную окружения *PATH*:

- **Use Git from Git Bash only** - способ установки, при котором переменная *PATH* не затрагивается совсем. Запуск **Git** в этом случае будет возможен только командами контекстного меню «*Git Bash Here*» и «*Git GUI Here*». Данный способ не рекомендуется, так как может потребовать дополнительной работы по конфигурации после установки **Git**;

- **Use Git from the Windows Command Prompt** - значение, предлагаемое по умолчанию. При выборе данной опции программа установки вносит минимальные изменения в *PATH*, прописывая только путь к консольной версии, чтобы не загромождать окружение инструментарием Unix. Команду контекстного меню «*Git Bash Here*» также можно



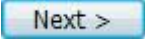
использовать. Рекомендуется

использовать данный вариант

запуска Git;

- **Use Git and optional Unix tools from the Windows Command Prompt** - программа установки вносит бóльшие изменения в *PATH*, чем в предыдущем варианте, прописывая пути к различным утилитам Unix.

**Предупреждение:** при выборе данной способа вместо Windows-утилит (таких, например, как *find* и *sort*) будут вызываться Unix-утилиты с аналогичными именами. Выбирайте данный способ только если вы полностью осознаёте, к какому результату он приведёт.

Установите переключатель в нужное положение и щёлкните по кнопке  для продолжения установки.

Откроется окно, в котором вам нужно установить способ обработки конца строк в текстовых файлах. Команды перевода строк в Unix-системах и в не Unix-системах различаются. Если в Unix-системах для «перевода строки» и «возврата каретки» используется команда LF (*line feed* - «подача бумаги на строку»), то в не Unix-системах используются две отдельные команды: LF+CR (*line feed + carriage return* - «возврат каретки»). Поэтому, на данном шаге требуется указать программе-установщику, файлы какого типа у вас используются и как **Git** должен обрабатывать окончание строк в текстовых файлах.

Здесь также возможны три варианта:

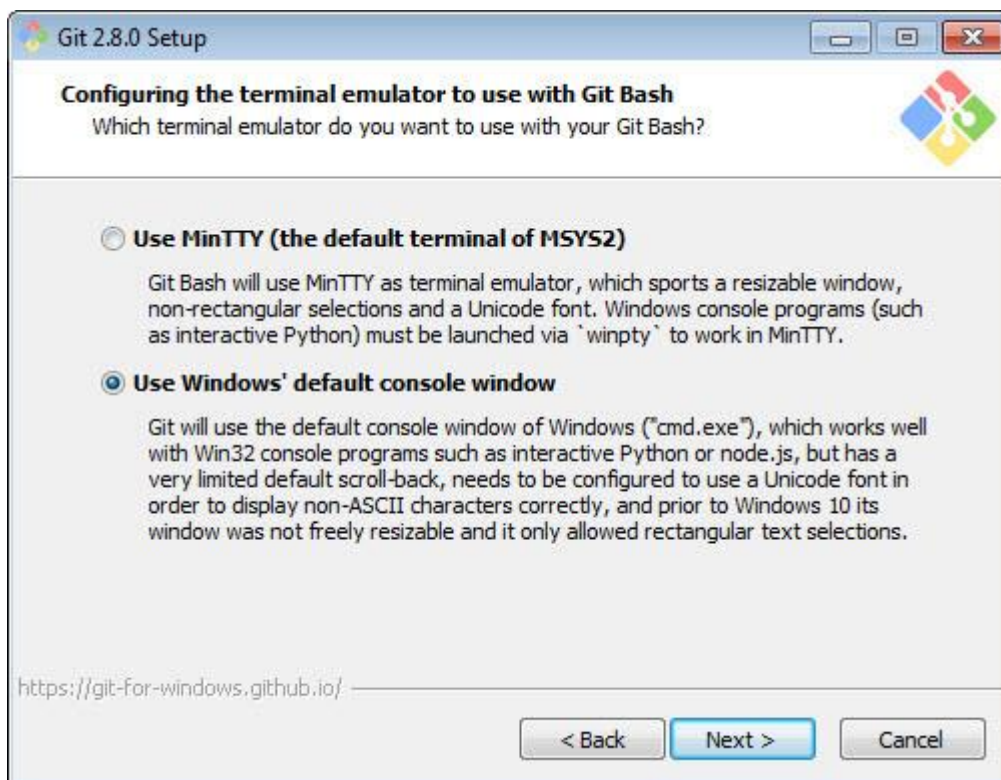
- **Checkout Windows-style, commit Unix-style line ending** - значение по умолчанию. При выборе данного способа **Git** будет конвертировать LF в CRLF при проверке текстовых файлов, а во время фиксирования изменений (коммитов) CRLF будет конвертироваться в LF. Данный способ рекомендуется для Windows-систем при работе с кросс-платформенными проектами (настройка проверки `core.autocrlf` установлена в `true`);

- **Checkout as-is, commit Unix-style line ending** - при установке переключателя в это положение **Git** не будет проводить конвертацию при проверке текстовых файлов. Во время коммитов CRLF будет конвертироваться в LF. Данный способ рекомендуется для Unix-систем при работе с кросс-платформенными проектами (настройка проверки `core.autocrlf` установлена в `input`);

○ **Checkout as-is, commit as-is** - при установке переключателя в данное положение конвертирование производиться не будет. Выбор данного варианта не рекомендуется для кросс-платформенных проектов (настройка проверки `core.autocrlf` установлена в `false`).

Установите переключатель в нужное положение и щёлкните по кнопке **Next >** для продолжения установки.

Откроется окно для выбора эмулятора терминала, который будет использоваться с **Git Bash**.

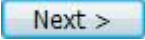


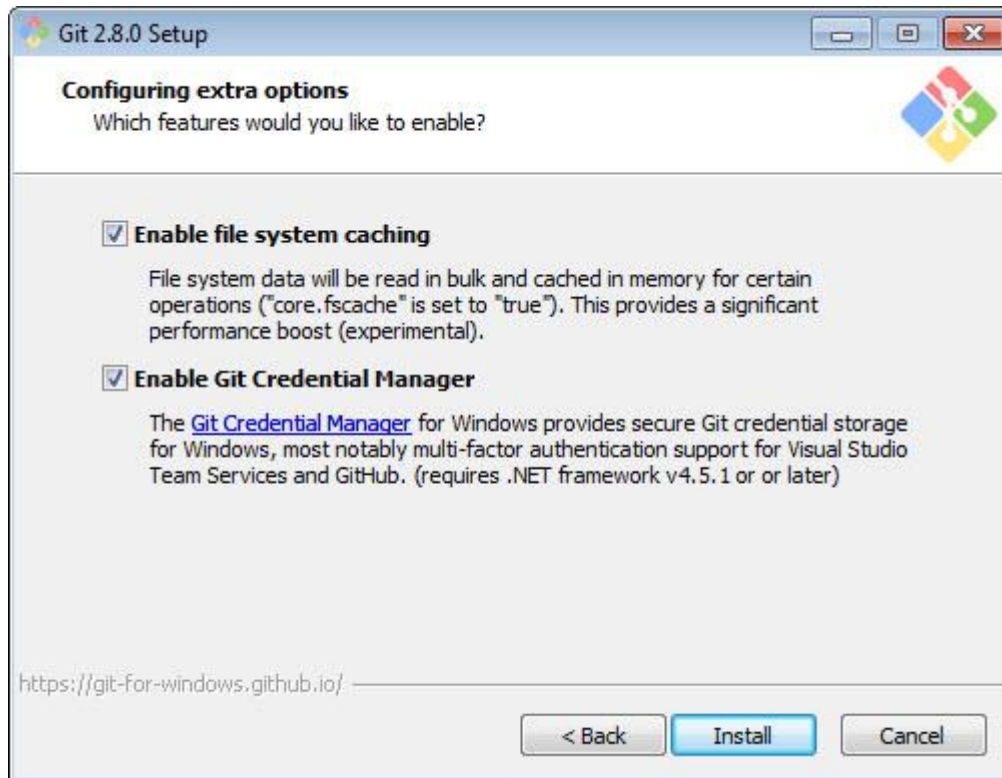
В качестве эмулятора терминала можно использовать:

○ **Use MinTTY (the default terminal of MSYS2)** - при установке переключателя в это положение **Git Bash** будет использовать MinTTY в качестве эмулятора терминала. MinTTY поддерживает окно с изменяемыми размерами, выделение непрямоугольной области и шрифт Unicode. Консольные Windows-программы (такие как интерактивный *Python*), должны запускаться через `winpty` для работы в MinTTY;

○ **Use Windows' default console window** - при выборе этого варианта **Git** будет использовать консоль Windows (`cmd.exe`) в качестве консоли по умолчанию. Она хорошо работает с Win32-консольными программами, такими как интерактивная *Python* или *node.js*, но имеет очень ограниченную прокрутку назад для того, чтобы быть сконфигурированной для использования шрифта Unicode с целью корректного отображения не ASCII-символов.


При работе с Windows 10, консольное окно не имеет свободно изменяемых размеров и позволяет только выделение прямоугольной области текста. Рекомендуется установить данный эмулятор терминала.

Установите переключатель в нужное положение и щёлкните по кнопке  для продолжения установки.

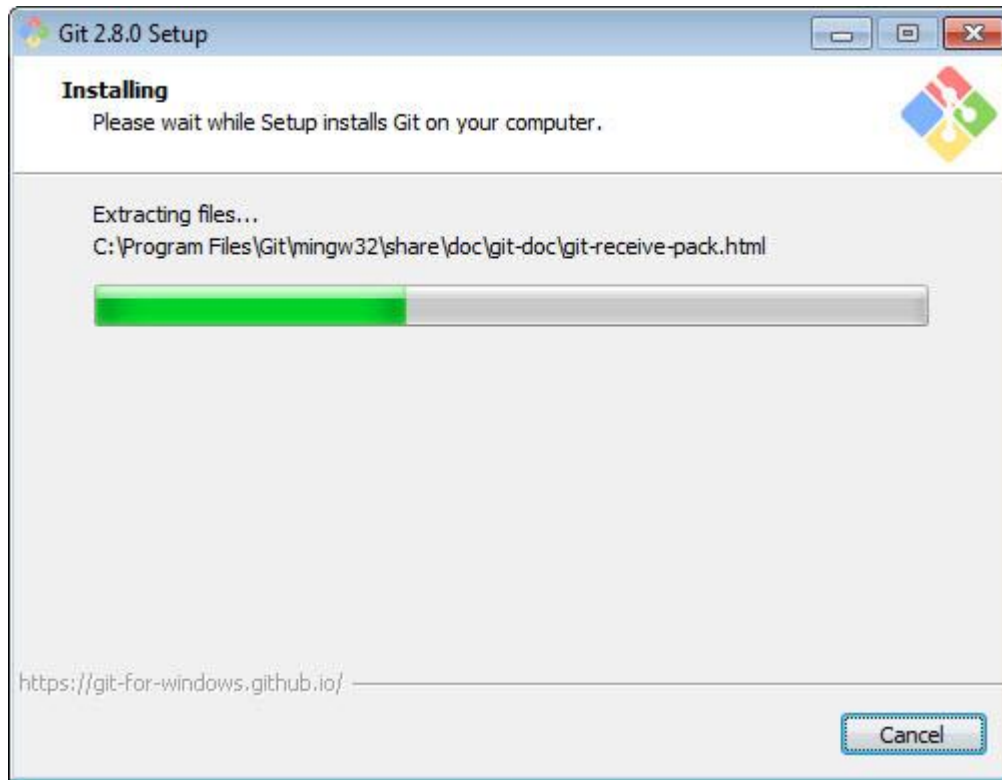


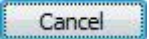
Откроется окно для установки дополнительных опций.

Дополнительными опциями являются:

- **Enable file system caching** - если опция установлена, данные файловой системы будут прочитаны все вместе и заэкшированы в память для выполнения определённых операций. Это обеспечивает значительный прирост производительности (экспериментальные данные);
  - **Enable Git Credential Manager** - [Git Credential Manager](#) для Windows обеспечивает безопасное хранение учётных данных **Git** для Windows, в особенности поддержку многофакторной аутентификации для служб Visual Studio и GitHub (должна быть установлена .NET framework версии 4.5.1 или более поздняя).
- Установите флаги у тех дополнительных опций, которые вы хотите подключить к **Git**, и щёлкните по кнопке  для запуска непосредственно процедуры установки.

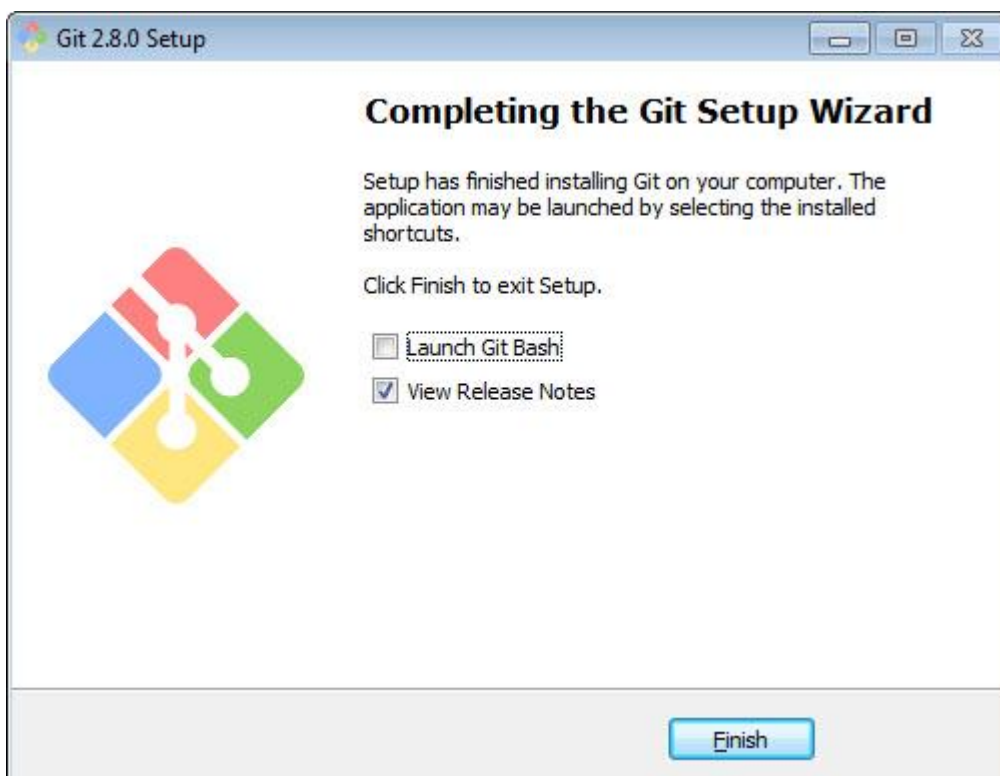
Откроется окно копирования файлов на ваш компьютер.



Прервать процесс копирования и отменить установку вы можете, щёлкнув по кнопке .

Дождитесь окончания копирования.

Откроется окно завершения установки:




В окне вы можете установить флаги: ○ **Launch Git Bash** - запустить **Git Bash** сразу по



завершению установки;

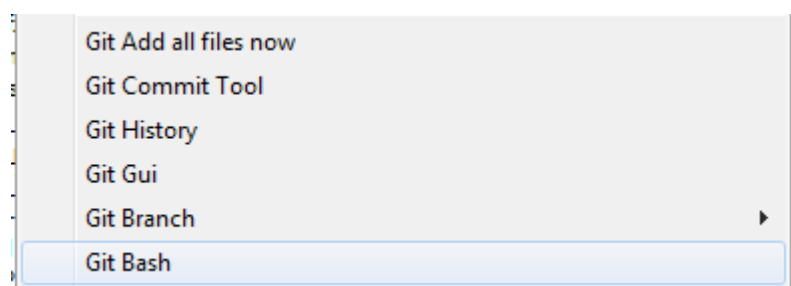
○ **View Release Notes** - просмотр файла

с описанием релизов **Git** для Windows.

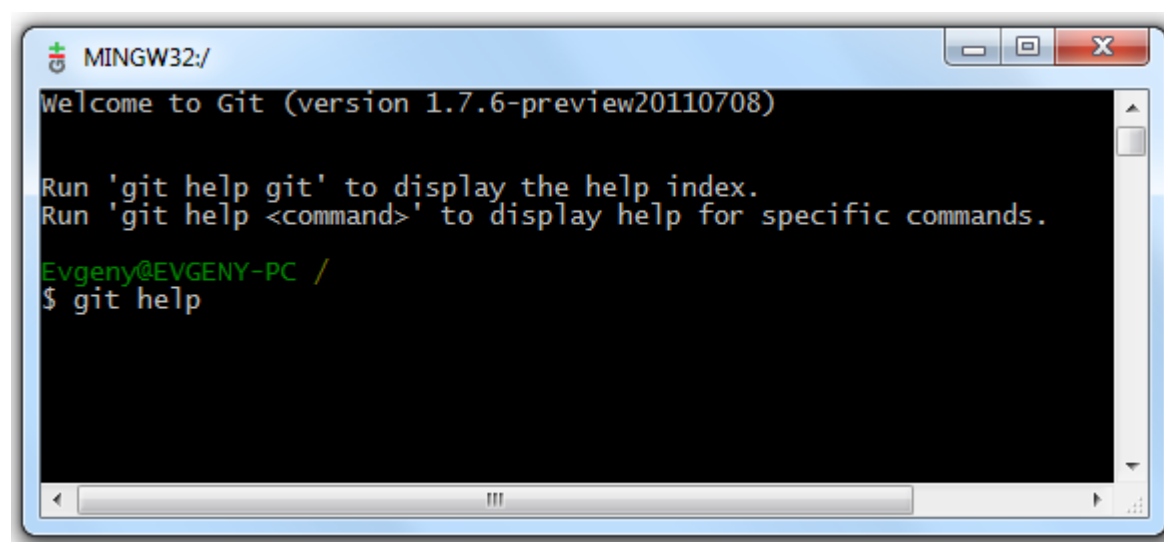
Для завершения установки щёлкните по кнопке . Программа установки завершит свою работу.

На этом установка **Git** закончена.

После установки выбираем в контекстном меню Проводника Git Bash:



или через Git Bash.lnk в папке с установленной программой:



Прописываем в консоли свои данные и настройки переносов строк: `git config --global user.name "ваше имя" git config --global user.email "ваша почта" git config --global core.autocrlf true git config --global core.safecrlf true`

### 13. Практическое занятие 7. Использование RAD технологии для прототипирования.

*Рассматриваются способы организации главного меню, всплывающего меню и панели инструментов, рассматривается компонент TImageList и возможность вывести изображение на пункты меню и кнопки панели.*

#### Цель занятия

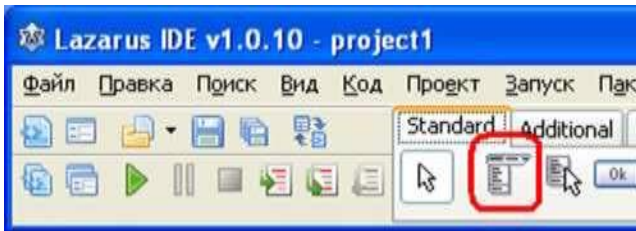
Практически отработать методику отладки приложения, содержащего главное и всплывающее меню, панели инструментов, изучение компонента **TImageList**.

#### Задание:

1. Отладить оконное приложение согласно прилагаемой ниже методике, применяя отличные от предлагаемых стили и дизайн. Предусмотреть обработку ввода ошибочной информации. Приветствуется усовершенствование приложения.
2. Адаптируйте готовое приложение применительно к одной из функциональных задач или одному рабочему месту Вашего учебного проекта.
3. В качестве отчёта представить архив с проектом (без \*.exe файла) и файлом отчёта, в котором описать отличия разработанного приложения от приложения, описанного в данном методическом пособии.

#### Главное меню

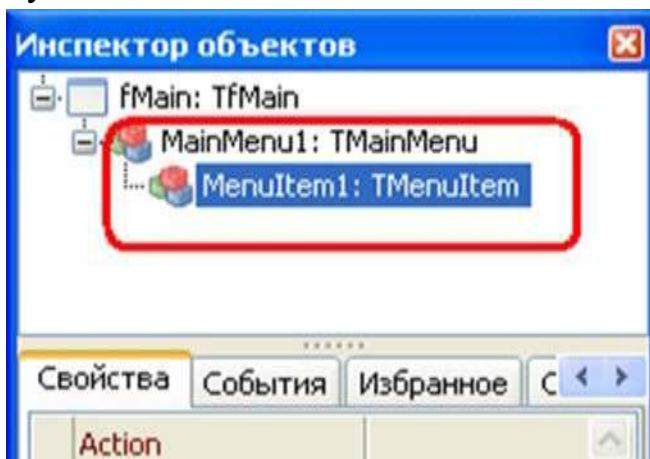
Интерфейс программы может содержать два типа меню: главное и всплывающее. Главное меню представляет собой строку в верхней части экрана. Первой командой меню, как правило, является "**Файл**", последней - "**Справка**". Реализуется главное меню компонентом **TMainMenu**, который находится на вкладке **Standard Палитры компонентов**:



**Рис. 13.1.** Компонент TMainMenu

Но давайте знакомиться с этим компонентом сразу на практике. Итак, откройте **Lazarus** с новым проектом. Сразу же переименуйте главную форму в **fMain**, проект сохраните в папку **13-01** под именем **ImageViewer**, а модулю главной формы дайте имя **Main**. В **Caption** формы напишите **Просмотр изображений**. А саму форму сделайте побольше, хотя бы 450\*700 пикселей.

Теперь установите на форму компонент **TMainMenu** - компонент невидимый, поэтому его можно установить в любой место, например, прямо посередине формы. Нам не придется обращаться к компоненту по имени, так что имя можно не изменять. Чтобы создать меню, нужно дважды щелкнуть по компоненту, и откроется **Редактор меню**. По умолчанию, в **Инспекторе объектов** окажется выделенным первый (и единственный) пункт меню **MenuItem1**:



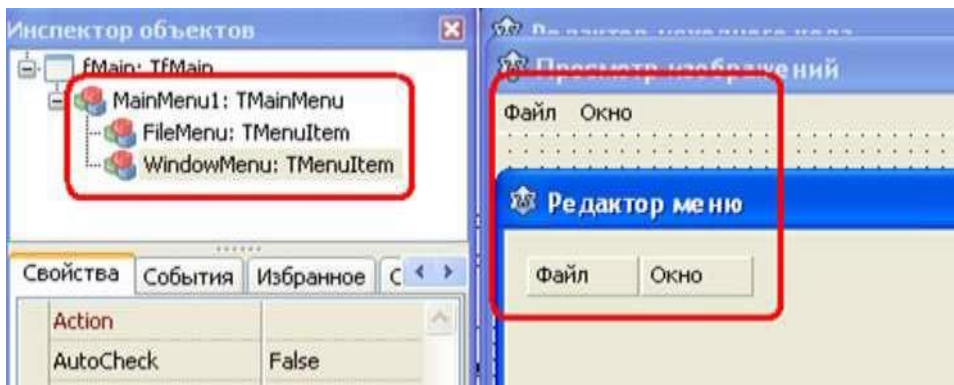
**Рис. 13.2.** Выделен пункт меню

Вот тут нам придется изменить некоторые настройки, так как имя **MenuItem1** нам ни о чем не говорит. Свойство **Name** переименуйте в **FileMenu**, в свойстве **Caption** напишите **Файл**. Сразу же вы увидите, как изменилось название на кнопке в **Редакторе меню**, и слово "Файл" появилось в верхней части окна формы. Щелкать по нему не нужно, иначе

будет создан обработчик нажатия на пункт меню, а нам это не нужно - для пункта "Файл" обработчик не предусмотрен.

Далее, в **Редакторе меню** щелкните по кнопке "**Файл**" правой кнопкой мыши, и в открывшемся всплывающем меню выберите команду "**Вставить новый пункт (после)**". В **Редакторе меню** появилась вторая кнопочка с надписью "**New Item2**". Её придется переименовать, как и первую - убедитесь, что именно она выделена в

**Инспекторе объектов**, переименуйте **Name** в **WindowMenu**, а в **Caption** напишите **Окно**. Теперь наше меню содержит два пункта: "**Файл**" и "**Окно**":



**Рис. 13.3.** Установка пунктов меню

Научимся создавать подпункты меню. Щелкните правой кнопкой мыши по кнопке "**Файл**" в **Редакторе меню** и выберите команду "**Создать подменю**". Сразу же на кнопке "**Файл**" появилась стрелка вниз, указывающая, что пункт меню содержит подпункты. И сам подпункт - кнопочку с надписью "**New Item3**", расположенную ниже "**Файл**". Убедимся, что в **Инспекторе объектов** выделен этот пункт, **Name** изменим на **FileOpen**, **Caption** - на **Открыть**.

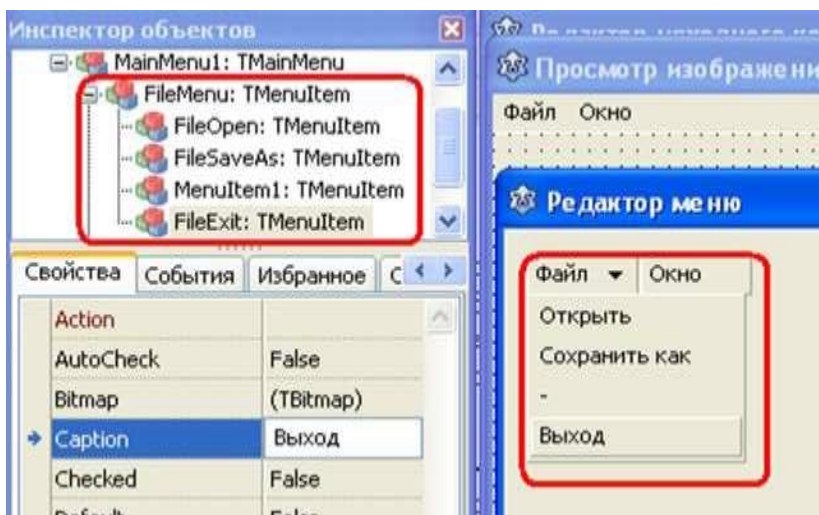
Тут идея такова: главные пункты меню называть, как **FileMenu**, **WindowMenu**, **OptionsMenu**, **HelpMenu** и т.п. Последнее - **Menu** будет указывать на то, что это главный пункт. Подменю именовать иначе - вначале идет имя главного пункта, затем команда. Например, для главного пункта **FileMenu** мы могли бы создать такие команды, как **FileOpen**, **FileSaveAs**, **FileClose** и т.п. Это лишь рекомендация, правила

вы можете придумать и свои, главное, чтобы вы не запутались в названиях главного пункта и его подпунктов.

Щелкните правой кнопкой мыши по кнопке **"Открыть"**, выберите команду **"Вставить новый пункт (после)"**. Полученный пункт переименуйте в **FileSaveAs**, а в **Caption** напишите **Сохранить как**.

Ниже аналогичным образом вставьте новый подпункт. У этого подпункта имя мы менять не будем, а в свойстве **Caption** мы впишем всего один символ **"-"**. Таким образом в меню вставляются разделители - горизонтальные черточки между командами.

Далее вставим еще один подпункт. Назовите его **FileExit**, а в **Caption** напишите **Выход**. В результате у вас должно получиться нечто подобное:



**Рис. 13.4.** Меню "Файл"

У нас ещё остался необработанным пункт **"Окно"**. Выделите его и создайте следующие подпункты с парой свойств (**Name** - **Caption**): **WindowNormal** - Нормальное, **WindowMinimize** - Свернуть, **WindowMaximize** - Развернуть. Как только вы это сделаете, можете закрывать **Редактор меню** - он нам больше не нужен.

В центральную часть окна поместите компонент **TImage**, в который мы и будем загружать выбранные пользователем изображения. Настройте компонент следующим образом:

- **Align** = **alClient**

- `Center = True`
- `Proportional = True`
- `Stretch = True`

Кроме того, на форме нам понадобятся два диалога: `TOpenPictureDialog`, который мы переименуем в `OPD`, и `TSavePictureDialog`, который назовем `SPD`. Как и `TMainMenu`, диалоги будут невидимы пользователю, поэтому их можно установить на любое место формы.

Выделите компонент `TOpenDialog`, разберемся с его свойствами. Как видите, их не очень много, да и нужны нам будут не все.

**DefaultExt**- расширение имени файла по умолчанию. В зависимости от

|                          |  |
|--------------------------|--|
| <p><b>FileName</b></p>   | <p>того, с каким типом файлов нам придется работать, такое у них будет и расширение. Если мы работаем с текстом, то лучше именам файлов давать расширение <code>txt</code>. Это необязательно, но так системе проще будет понять, с каким типом файлов ей придется работать.</p> <p>- имя файла. Можно сразу же указать имя файла, с адресом и расширением, но обычно это свойство оставляют пустым. После того, как диалог сработает, и пользователь выберет файл для открытия, то в этом свойстве будет и адрес, и имя этого файла. А они нам будут нужны в методе <code>Memo1.LoadFromFile()</code></p> |
| <p><b>Filter</b></p>     | <p>- фильтр типов файлов. Здесь можно задать фильтрацию файлов по их типам, используя маску файлов. В маске знак <code>"**"</code> означает любое количество любых символов. Например, в фильтре можно указать маски <code>*.txt</code>, что означает "любой файл с расширением <code>txt</code>", и(или) <code>*.*</code>, что означает "любой файл с любым расширением или без расширения".</p>  |
| <p><b>InitialDir</b></p> | <p>- папка (директория, каталог), используемая по умолчанию. Здесь указывается адрес папки, с которой диалог начнет свою работу. Заполнять это свойство имеет смысл лишь тогда, когда нужные файлы у вас будут храниться в каком-то одном, конкретном месте. В нашем случае мы не знаем, где пользователю захочется сохранять свои файлы, поэтому заполнять это свойство не нужно.</p>   |

|              |   |
|--------------|---|
| <b>Name</b>  | <p>- имя компонента. С этим свойством вы уже знакомы, но тут нужно сделать одно замечание. Помните, в прошлой лекции я говорил, что если компонент один, то его можно не переименовывать? С кнопками этот совет хорош - к кнопкам в коде нам не приходится обращаться по имени. А вот к диалогам придется обращаться неоднократно. Судите сами, что проще будет набрать на клавиатуре:</p> <p><b>OpenDialog1.FileName</b> или <b>OD.FileName</b>?</p> <p>Вот то-то. Поэтому совет: переименовывайте диалоги, оставляя лишь заглавные буквы имени. Если вам доведется в одной форме использовать два одинаковых диалога (а мне пока такого делать не доводилось), то можно их назвать <b>OD1</b> и <b>OD2</b>.</p> |
| <b>Title</b> | <p>- заголовок окна диалога. По умолчанию, он уже содержит нужный текст: "Открыть существующий файл". Если же вам захочется установить какое то свое, нестандартное название, например "Открыть мой текстовый файл", можете вписать его здесь.</p>  |

Со свойствами разобрались, остался метод. Метод этот - **Execute**.

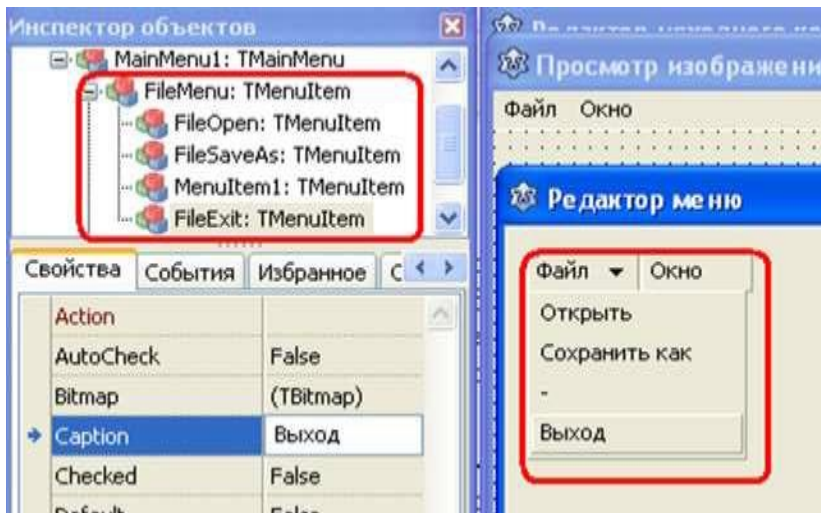
**Execute** вызывает стандартное диалоговое окно, и дает пользователю сделать выбор. Если пользователь сделал свой выбор, то **Execute** вернет **True**. Если пользователь отказался делать выбор и закрыл окно, не указав файла, метод вернет **False**.

Настроим наш **TOpenDialog**:

- **DefaultExt = .jpg**
- **Name = OD**

Теперь займемся фильтром. Выделите свойство **Filter** и щелкните по кнопке "...". справа от него. Откроется редактор фильтров, в нем надо поставить фильтры, обеспечивающие нам открытие файлов с изображениями, вместо текстовых файлов поставьте соответствующие расширения:





**Рис. 13.4.1** Фильтры для TOpenDialog

Когда вы нажмете **"ОК"**, Lazarus в свойстве **Filter** сгенерирует следующий фильтр:

Текстовые файлы|\*.txt|Все файлы|\*.\*

Эту строку можно было бы написать и вручную, но воспользоваться **Редактором фильтров** все же удобнее.

Теперь займемся программированием пунктов меню. Выберите на форме пункт **"Файл -**

**> Выход"**, и автоматически будет создан обработчик **OnClick** для этого пункта. В обработчике впишите всего одну команду:

```
procedure TfMain.FileExitClick(Sender: TObject);
begin
    Close;
end;
```

Оператор **Close** в дочернем (неглавном) окне закрывает это окно, а в главном окне завершает работу программы. Создайте таким же образом



обработчик для **"Файл -> Открыть"**, с его кодом вы должны быть знакомы по предыдущему проекту:

```
procedure TfMain.FileOpenClick(Sender:
TObject); begin
    if OPD.Execute then
        Image1.Picture.LoadFromFile(OPD.FileName); end;
```

Для **"Файл -> Сохранить как"** код будет похожий:

```
procedure TfMain.FileSaveAsClick(Sender:
TObject); begin
    if SPD.Execute then
        Image1.Picture.SaveToFile(SPD.FileName); end;
```

С пунктом **"Файл"** разобрались, остался пункт **"Окно"** и три его подпункта. Здесь мы будем менять состояние окна у **fMain.WindowState**. Код для этих событий следующий:

```
procedure TfMain.WindowMaximizeClick(Sender:
TObject); begin
    fMain.WindowState:=
wsMaximized; end;
```

```
procedure TfMain.WindowMinimizeClick(Sender:
TObject); begin
    fMain.WindowState:=
wsMinimized; end;
```

```
procedure TfMain.WindowNormalClick(Sender:
TObject); begin
    fMain.WindowState:=
wsNormal; end;
```

Не перепутайте названия событий. Сохраните проект, запустите его и попробуйте в работе. Программа должна нормально загружать файлы, менять состояние окна,

завершать работу. При сохранении файла указывайте не только желаемое имя, но и расширение, например *newfile.jpg* - компонент **TImage** не умеет угадывать формат и самостоятельно подставлять нужное расширение.

### Всплывающее меню

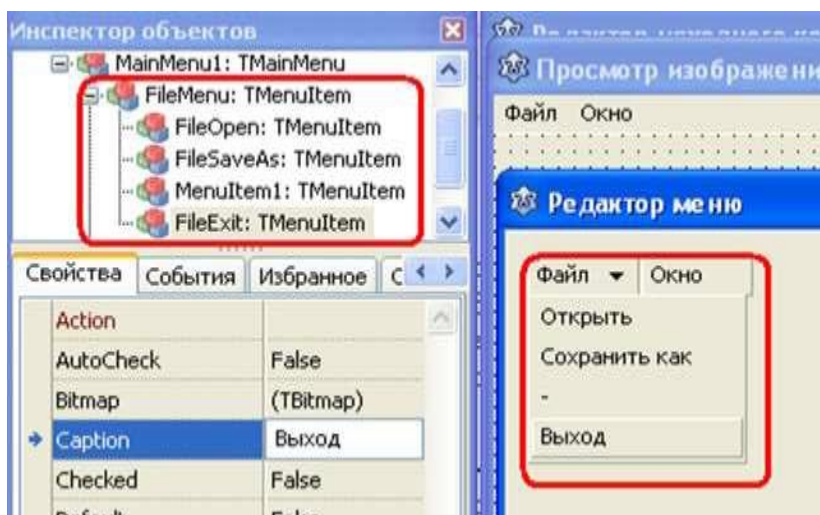
Мы с вами то и дело пользуемся всплывающим меню. Это меню, которое появляется, когда пользователь нажимает правую кнопку мыши. Реализуется оно компонентом **TPopupMenu**, который находится рядом с **TMainMenu** на **Палитре компонентов**.

Поскольку оно также невидуально, устанавливайте его на любое место на форме.

Редактор всплывающего меню также вызывается двойным нажатием на компонент (или нажатием на "..." в свойстве **Items**). Редактируется оно примерно также. Выделите первый (и пока единственный) пункт этого меню, в свойстве **Name** напишите **pFileMenu**, в свойстве **Caption** напишите **Файл**. Далее, щелкните правой кнопкой мыши по кнопке этого пункта в **Редакторе меню**, и выберите **"Создать подменю"**. Создайте тут два подраздела: **pFileOpen** и **pFileSaveAs** со свойствами **Caption**, соответственно, **Открыть** и **Сохранить как**.

Аналогичным образом, создайте пункт **pWindowMenu** (**Caption** = **Окно**) с подпунктами **pWindowNormal**, **pWindowMinimize** и **pWindowMaximize** (**Caption** = **Нормальное**, **Свернуть**, **Развернуть**).

Далее, создайте главный пункт с **Caption** = **"-"** (имя не меняем, это просто разделитель), и еще один пункт **Name** = **CloseMenu**, **Caption** = **Выход**. У

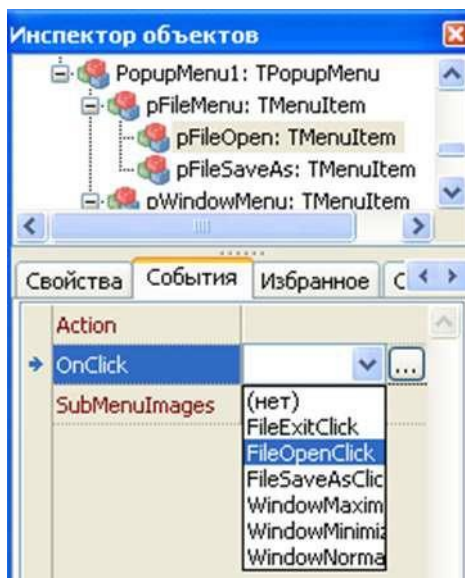


вас должно было получиться такое меню:

**Рис. 13.5.** Всплывающее меню

Меню мы сделали, но не спешите закрывать **Редактор меню**. Если вы думаете, что для всплывающего меню мы снова будем писать тот же самый код, то вы ошибаетесь, все намного проще. Выделите подпункт "**Файл -> Открыть**" - щелкайте один раз, а не дважды, иначе будет создан обработчик **OnClick**, а нам это не нужно. Если это всё же случилось, вам придется удалить пустой каркас процедуры этого обработчика, а также объявление этой процедуры в разделе **type** в верхней части модуля формы. Только не перепутайте названия процедур - все пункты всплывающего меню у нас начинаются с буквы **p**.

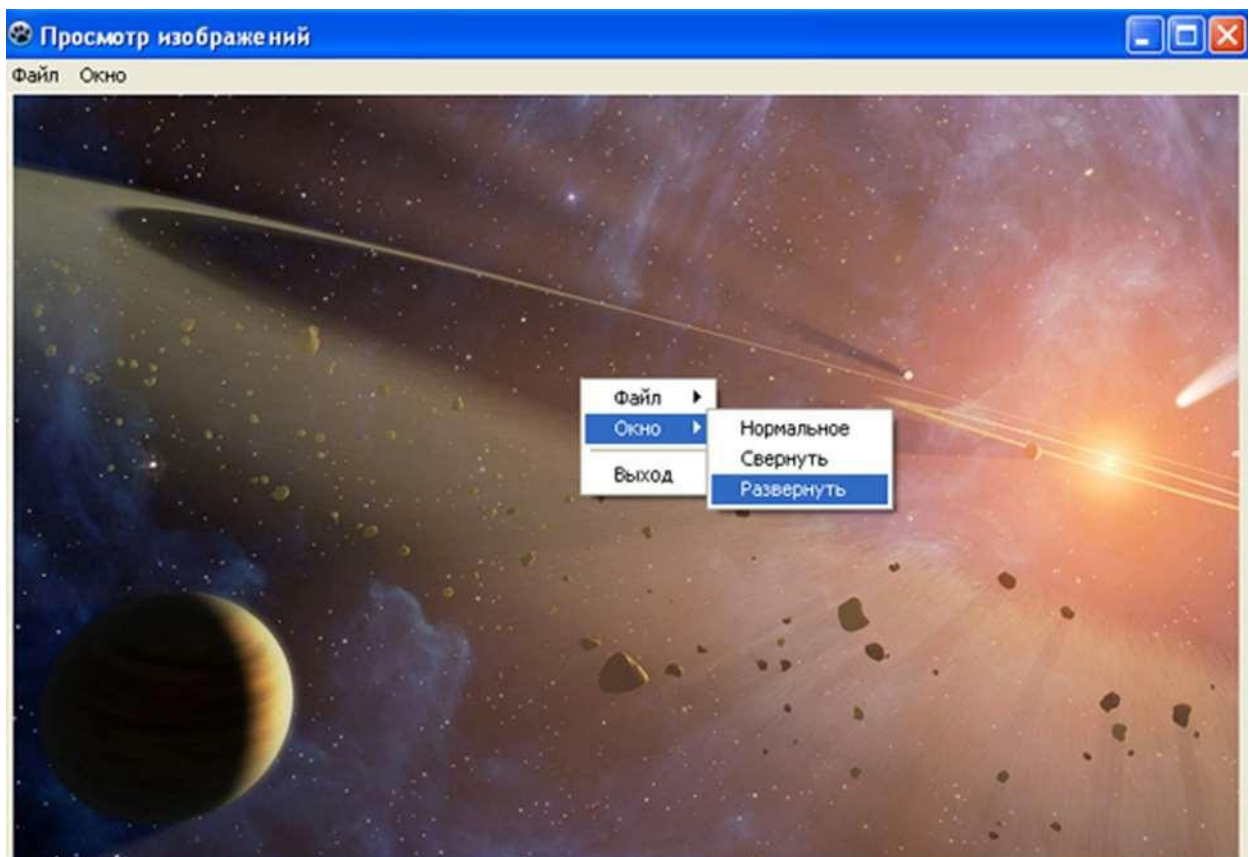
Итак, мы выделили подпункт всплывающего меню "**Файл -> Открыть**". В **Инспекторе объектов** перейдите на вкладку **События**, откройте список в событие **OnClick** и выберите из этого списка событие **FileOpenClick** - это событие открытия файла в главном меню, которое мы ранее запрограммировали:

**Рис. 13.13.** Выбор готового события в OnClick

Таким образом, когда пользователь выберет эту команду в всплывающем меню, будет вызвано соответствующее событие главного меню, так что нам не придется дважды писать один и тот же код. Выберите соответствующие события главного меню для оставшихся команд всплывающего меню "**Файл -> Сохранить как**", "**Окно -> Нормальное**", "**Окно -> Свернуть**", "**Окно -> Развернуть**" и "**Выход**". После этого можно закрыть **Редактор меню**.

Однако и этого недостаточно, чтобы пользователь мог вызвать

всплывающее меню - его еще нужно привязать к какому-нибудь компоненту. Обычно его привязывают либо к форме, либо к панели. Если на форме две панели, то для каждой можно сделать своё всплывающее меню, со своими командами. У нас всю часть формы занимает компонент **Image1**, к нему и привяжем всплывающее меню. Выделите **Image1**, перейдите на вкладку **Свойства Инспектора объектов**, и в свойстве **PopupMenu** выберите наш **PopupMenu1**. Теперь, если пользователь нажмет над **Image1** правую кнопку мыши, выйдет всплывающее меню. Если вы все сделали правильно, то команды всплывающего меню будут вызывать соответствующие команды главного



меню, и все будет работать нормально:

### увеличить изображение

**Рис. 13.7.** Работа всплывающего меню в загруженном приложении

### Компонент TImageList

**TImageList** - это контейнер для хранения списка изображений. Как правило, в **TImageList** хранят изображения для меню и кнопок панелей инструментов. Компонент **TImageList** находится на вкладке **Common Controls** **Палитры компонентов**:



**Рис. 13.8.** Компонент TImageList

Компонент не визуальный, так что его также можно установить на любое свободное место. Щелкните дважды по нему, чтобы открыть **Редактор ImageList**. В Редакторе нажмите кнопку "**Добавить**", откроется диалоговое окно "**Добавить изображения**". Теперь вот что: для пунктов меню нам нужны маленькие изображения, 16\*16 пикселей. По умолчанию, Lazarus устанавливается в папку **C:\Lazarus**. Если вы не меняли эту папку, то картинки для пунктов меню у вас будут по адресу

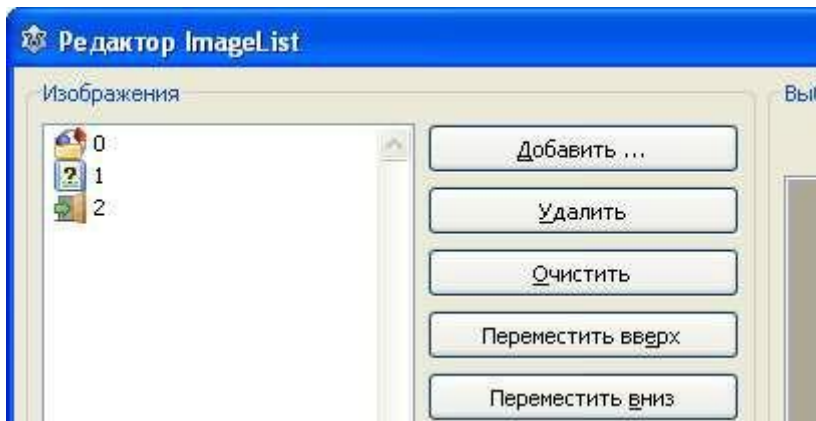
**C:\Lazarus\images\menu**

Изображений там не так, чтобы очень много, гораздо больше интересных изображений, да и целых бесплатных коллекций изображений можно найти на просторах Интернета. Однако на первое время обойдемся тем, что уже есть. Откройте указанный адрес в диалоге выбора изображений. Нам нужна картинка для пункта меню "**Файл -> Открыть**". Тут лучше подойдет файл **menu\_project\_open.png**. Выберите этот файл и нажмите кнопку "**Открыть**" - файл попадет в список под индексом 0. Как и в других списках, индексация в **TImageList** начинается с 0, индекс -1 означает, что изображения не выбрано. Снова нажмите кнопку "**Добавить**" в редакторе. Для "**Файл -**

**> Сохранить как**" подойдет файл **menu\_project\_saveas.png**. Изображение встанет под индексом 1.

Впрочем, изображения можно перемещать по списку, менять их местами с помощью кнопок "**Переместить вверх**" и "**Переместить вниз**". Под индексом 2 установите файл **menu\_exit.png** - это будет изображение для "**Файл -> Выход**". Для пунктов меню "**Окно**" подходящих изображений нет, поэтому мы оставим эти команды без соответствующих картинок.

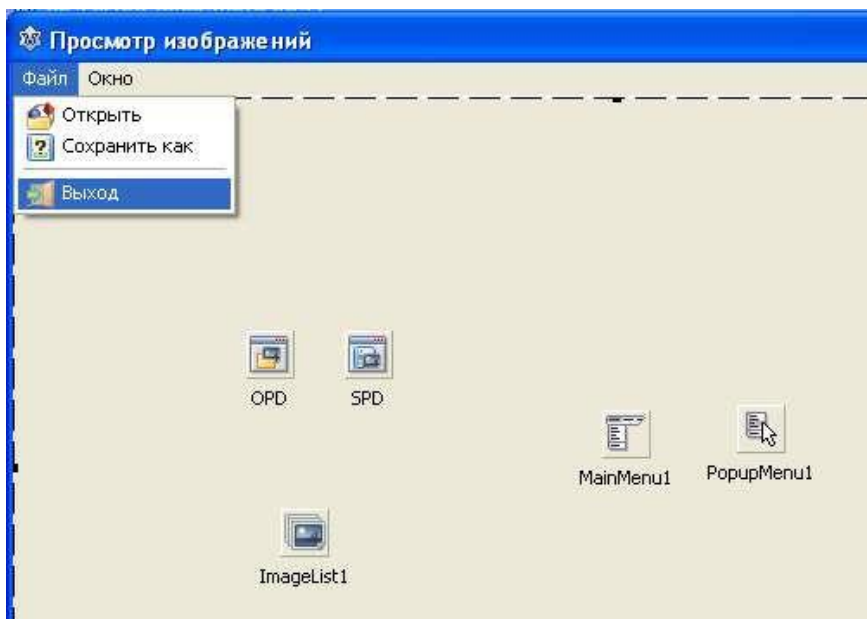




**Рис. 13.9.** Три установленных изображения в редакторе ImageList

Нажмите кнопку **"ОК"**, чтобы закрыть редактор. Теперь нам нужно, чтобы главное меню увидело эти изображения. Выделите компонент **MainMenu1**. В его свойстве **Images** выберите наш **ImageList1**. Таким образом, мы связали главное меню со списком изображений.

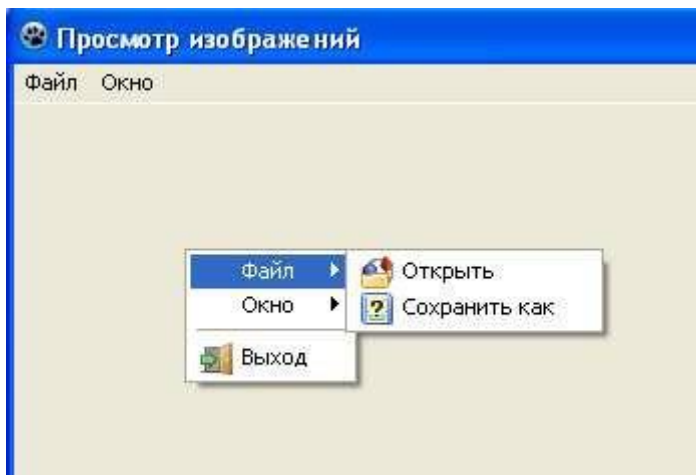
Далее, дважды щелкните по **MainMenu1**, чтобы открыть редактор меню. Выделите подпункт **"Открыть"** пункта **"Файл"**. Вы видите, что его свойство **ImageIndex** равно -1, то есть, изображения не выбрано. Откройте список изображений и выберите первое изображение, с индексом 0. Для подпункта **"Сохранить как"** установите индекс 1. Для **"Выход"** - индекс 2. В результате пункты меню **"Файл"** станут отображаться вместе с изображениями:



**Рис. 13.10.** Меню с изображениями

Большой пользы от этого нет, но проект украшает. То же самое сделайте для соответствующих пунктов всплывающего меню: сначала в его свойстве **Images** выберите наш список **ImageList1**. Затем откройте редактор меню, выделите поочередно пункты "Открыть", "Сохранить как" и "Выход", выберите для них соответствующие **ImageIndex**. Сразу же в редакторе меню вы не заметите изменений, однако закройте его, сохраните проект и запустите программу на выполнение.

Щелкните правой кнопкой мыши по окну и, пожалуйста, меню с изображениями:



**Рис. 13.11.** Всплывающее меню с изображениями

### Горячие клавиши

Тут самое время вспомнить об одном важном свойстве подпунктов меню, как главного, так и всплывающего - **ShortCut**.

**ShortCut** - свойство, позволяющее выбрать для команды меню сочетание клавиш, так называемые "горячие клавиши". Если какое то сочетание клавиш присвоено одной из команд, то пользователю необязательно будет лезть в меню, чтобы выполнить эту команду, достаточно будет нажать соответствующее сочетание клавиш. Так, для

команды меню **Файл->Открыть** традиционно применяют сочетание **<Ctrl+O>**; для команды **Файл->Сохранить** - **<Ctrl+S>**; для команды **Файл->Выход** можно использовать **<Ctrl+Q>**. Откройте редактор главного меню, выделяйте по очереди в меню **Файл** подпункты "Открыть", "Сохранить как" и "Выход". Выберите для этих команд соответствующие сочетания в свойстве **ShortCut**. В результате меню приобретет такой вид:



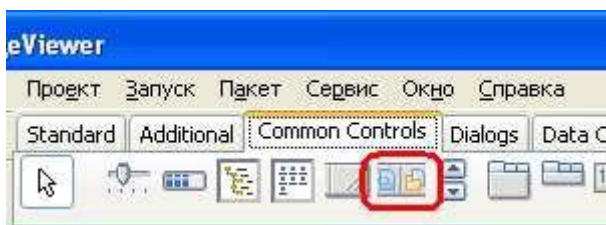
**Рис. 13.12.** Горячие клавиши в меню

Теперь пользователь сможет выполнять эти команды как с помощью меню, так и с помощью "горячих клавиш". Делать то же самое с пунктами всплывающего меню имеет смысл только в том случае, если его команды отличаются от команд главного меню. В нашем случае все равно будут выполняться команды главного меню, так что указывать эти же горячие клавиши во всплывающем меню не нужно.

### Панель инструментов

Самый простой способ организовать **Панель инструментов** - установить обычную панель **TPanel** с настройками свойств **Align = alTop**, **AutoSize = True**, и уже на эту панель установить кнопки **TSpeedButton**. И панель, и кнопки мы с вами изучали в [лекции №3](#), поэтому останавливаться на этом не будем. Изучим другой, более профессиональный способ организации панели инструментов.

Для этого нам потребуется компонент **TToolBar** с вкладки **Common Controls** **Палитры компонентов**:



**Рис. 13.13.** Компонент TToolBar

Просто установите его на форму, и он сам займет нужное место - растянется по всей верхней границе окна. Обращу ваше внимание на некоторые свойства компонента, которые могут представлять для нас интерес.



|                |   |
|----------------|---|
| Images         | - ссылка на список изображений для кнопок панели. Выполняет такую же функцию, как соответствующее свойство меню.  |
| DisabledImages | - ссылка на список изображений, которые будут привязаны к неактивным кнопкам. Чаще всего это свойство оставляют пустым, в таком случае у неактивных кнопок (у которых <b>Enabled = False</b> ) будет такое же изображение, как у активных, но только серого цвета.  |
| HotImages      | При желании для неактивных кнопок можно подобрать альтернативные изображения.<br>- ссылка на список изображений для кнопок, над которыми в данный момент находится указатель мыши. Чаще всего это свойство не заполняют, и тогда  |
| EdgeBorders    | изображения у таких кнопок не меняется. При желании, можно подобрать альтернативные изображения, чтобы выделять текущую кнопку.<br>- установка бордюра по сторонам панели. Имеет 4 подсвойства, соответствующие границам панели, и если в соответствующем подсвойстве установлено <b>True</b> , то  |
| Flat           | данная граница будет выделена. По умолчанию, включена только верхняя граница, отделяющая панель инструментов от строки с главным меню. Если желаете, можете включить отображение всех границ.<br>- если <b>True</b> , то кнопки плоские, как в современных панелях Windows, если <b>False</b> - выпуклые, как в старых программах. По умолчанию равно <b>True</b> , так лучше и оставить. |
| ShowHint       | - если <b>True</b> , то всплывающие подсказки будут выходить, если <b>False</b> , то нет. С таким свойством нам уже доводилось встречаться. На панели инструментов обычно отображаются кнопки с изображениями, но без текста.   |

Пользователь, если только знакомится с вашей программой, поначалу может и не сообразить, какая кнопка для чего нужна. Поэтому в панели инструментов данную подсказку лучше включать (по умолчанию **ShowHint = False**). Затем в свойстве **Hint** кнопок вы пропишите нужные подсказки, и пользователь сможет их прочесть, когда подведет указатель мыши к той или иной кнопке.

Сделаем на панели **ToolBar1** следующие установки:

|        |   |
|--------|---|
| EdgeBo | - включаем все границы, если есть желание. А вообще это необязательно.  |
| rders  | - выберите наш <code>ImageList1</code> . Как видите, один и тот же список изображений мы используем и для главного меню, и для всплывающего, и для панели инструментов. |
| Images |   |

`howHint = True`

Остальные свойство можно не изменять.

Займемся созданием кнопок. Вначале сделаем кнопку **"Выход из программы"**, затем установим разделитель, затем кнопки **"Открыть файл с изображением"** и

**"Сохранить файл как"**. Для команд пункта **"Окно"** мы кнопок делать не будем, тем более, что у нас нет для них изображений. Итак, щелкните правой кнопкой мыши по панели инструментов, и выберите команду **"Новая кнопка"**. На панели инструментов появилась кнопка, которой Lazarus автоматически присвоил имя **ToolButton1**.

Измените ее свойство **Name** на `bExit`, чтобы название соответствовало действию. В свойстве **Hint** напишите следующую подсказку:

**Выход из программы**

А в свойстве **ImageIndex** выберите изображение с индексом 2. Теперь перейдем на вкладку **События Инспектора объектов**, и в свойстве **OnClick** выберем

`FileExitClick` - событие **OnClick** для пункта **"Выход"** главного меню. Как видите, для панели инструментов нам тоже не придется заново писать код.

Снова щелкните правой кнопкой по панели инструментов, и выберите команду **"Новый разделитель"**. Тем самым вы вставите разделитель - вертикальную черту между предыдущей кнопкой и следующими. Ничего с этим разделителем делать больше не нужно, разве что вам покажется, что разделитель слишком (или наоборот, недостаточно) широк. В этом случае выделите разделитель и измените его свойство **Width**.

Вставьте еще одну кнопку. Назовите ее `bOpen`, выберите для нее

изображение под индексом 0, в свойстве **Hint** напишите следующую подсказку:

**Открыть файл с изображением**

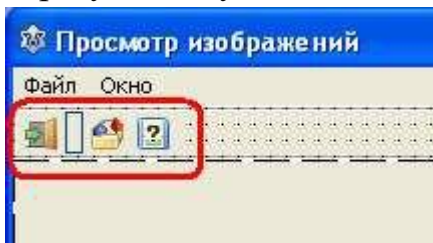
В событии **OnClick** этой кнопки выберите соответствующее событие из главного меню.

Вставим еще одну, последнюю кнопку. Назовем ее **bSaveAs**, в **ImageIndex** выберем картинку с индексом 1, в свойстве **Hint** напишем подсказку:

**Сохранить файл как**

Выберем для кнопки соответствующее событие **OnClick**.

В результате у вас должна получиться такая **Панель инструментов**:



**Рис. 13.14.** Панель инструментов в проекте

Это все. Сохраните проект и запустите его на выполнение. Если вы все сделали правильно, теперь у вас есть правильно действующая панель управления с тремя кнопками и одним разделителем.

Адаптируйте готовое приложение применительно к одной из функциональных задач или одному рабочему месту Вашего учебного проекта.

## Программирование на Lazarus

Ачкасов Вячеслав Юрьевич

<https://www.intuit.ru/studies/courses/13745/1221/inf>

О

## 14. Лабораторная работа 7. Создание прототипа ИС с использованием RAD технологий

### Лабораторная работа 16 . Телефонный справочник

*Данная лабораторная работа закрепляет пройденный материал по темам работы с записями, типизированными файлами и сеткой строк TStringGrid. В процессе работы подробно рассматривается создание программы - телефонного справочника.*

#### Цель занятия

Закрепление работы с записями, типизированными файлами и сеткой строк **TStringGrid**.

#### Задание

1. Отладить приложение согласно изложенной ниже методике
2. Включить приложение в виде модуля в свой проект

#### Постановка задачи

В данной лабораторной работе нам предстоит написать приложение - телефонный справочник. Это приложение должно обладать всеми необходимыми для нашей задачи качествами:

- выводить контактные данные в виде таблицы;
- сохранять контакты в файл и считывать их оттуда;
- добавлять новый или редактировать (изменять) существующий контакт;
- удалять выбранный из списка контакт;
- сортировать список контактов по алфавиту.

Для работы с контактами создадим тип данных запись, будем использовать переменную этого типа и типизированный файл, в котором данные будут храниться в двоичном виде. Сами контакты будем выводить в сетку **TStringGrid**, для добавления и редактирования записей создадим отдельное окно - сетка позволит только просматривать и выбирать контакты.

#### Реализация

Для начала откройте **Lazarus** с новым проектом. Форму, как обычно, называем **fMain**, сохраняем проект в папку **16-01** под именем **telephones**, модулю главной формы присваиваем имя **Main**. Пойдем дальше. Скорее всего, вы будете делиться этой программой с друзьями, поэтому имеет смысл сразу же отключить от проекта вставку отладочной информации. Если вы помните, нужно выбрать команду "**Проект -> Параметры проекта**", в разделе "**Параметры компилятора**" перейти на подраздел "**Компоновка**" и убрать флажок "**Генерировать отладочную информацию для GDB**". Это позволит нашей программе сразу "похудеть" - от 15 мегабайт готового продукта, до менее чем 2-х мегабайт.

Теперь займемся формой. В свойстве **Caption** напишите "**Телефонный справочник**", в **BorderStyle** выберите **bsDialog** (нам не нужно, чтобы пользователь менял размеры окна), а в свойстве **Position** - **poDesktopCenter**. Чтобы ваше приложение в точности соответствовало моему, выберите ширину формы **Width700** пикселей, а высоту **Height 400** пикселей.

Теперь нам на форме потребуется простая панель инструментов. Установите обычную панель **TPanel**, очистите её свойство **Caption**, в свойстве **Align** выберите **alTop**, чтобы панель заняла весь верх формы, и в свойстве **Height** установите значение 27 - нам не нужна слишком высокая панель инструментов, узкая будет смотреться аккуратней.

На панель нам нужно установить четыре кнопки **TSpeedButton** с вкладки **Additional Палитры компонентов**. Кнопки **TSpeedButton** нам нужны, чтобы на них не переходил фокус ввода, в этом случае, всегда активной будет только сетка. Присвойте свойству **Name** кнопок соответственно, имена **bAdd**, **bEdit**, **bDel** и **bSort**. Первая "**b**" в именах означает, что это кнопка - button. В свойствах **Caption** этих кнопок напишите, соответственно, "**Добавить контакт**", "**Редактировать контакт**", "**Удалить контакт**" и "**Сортировать список**" (без кавычек, разумеется). Картинок на кнопки мы ставить не будем, обойдемся только пояснительным текстом. Свойству **Top** всех кнопок присвойте значение 2, чтобы они почти прижались к верхней границе панели, свойству **Width** присвойте 150, чтобы весь текст помещался на кнопках без проблем. Переместите их относительно друг друга так, чтобы от левого края панели и друг от друга их отделяло тоже примерно по 2 пикселя.

Теперь сетка. На свободное место формы, ниже панели, установите сетку **TStringGrid**. Поскольку нам часто придется обращаться к ней в процессе кодирования, укоротим имя в свойстве **Name** до **SG**. В свойстве **Align** сетки выберите **alClient**, чтобы сетка заняла всё оставшееся место. Теперь нужно

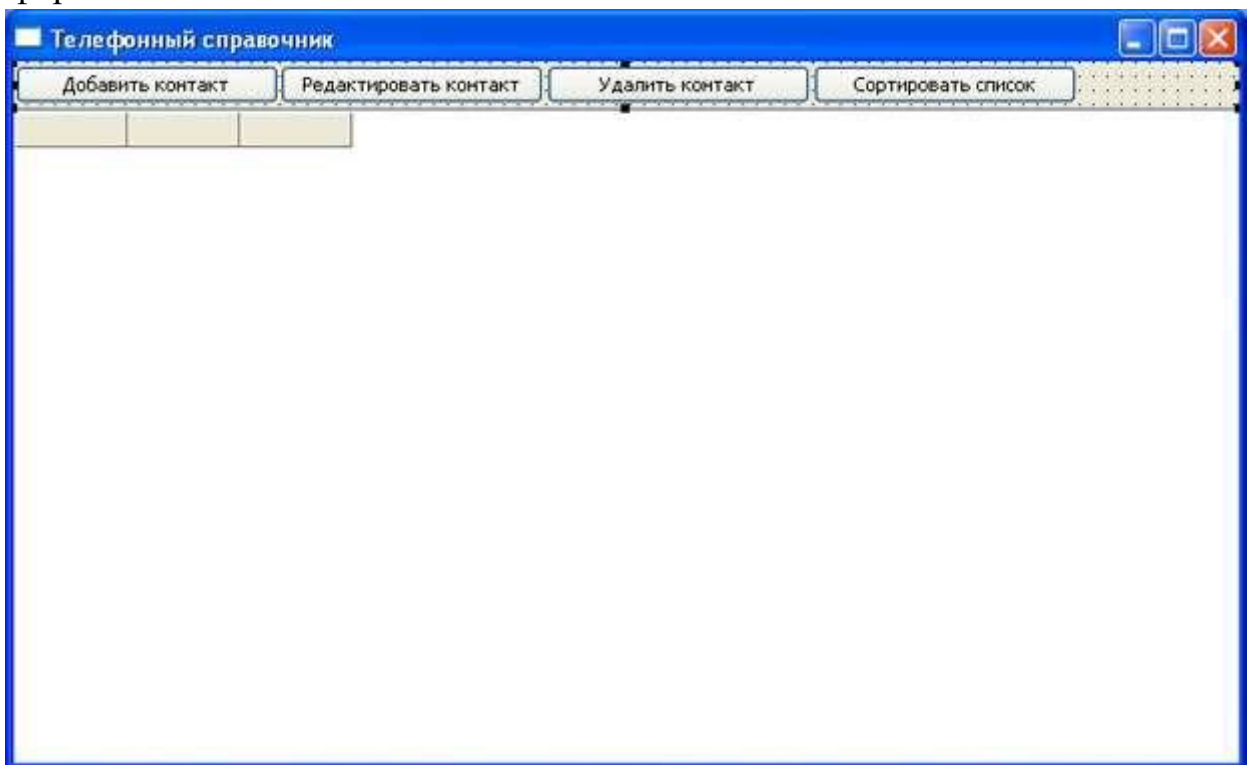
изменить некоторые другие настройки. По умолчанию, в сетке установлено по пять колонок **ColCount**, и по пять строк **RowCount**.

Колонок нам нужно только три: имя контакта, его телефон и примечание, какой это телефон (мобильный, домашний, служебный, соседский).

Поэтому уменьшим количество колонок **ColCount** до 3. А строка нам нужна и вовсе одна, она будет служить заголовком колонок, и она будет фиксированной. Уменьшайте **RowCount** до 1.

Теперь о фиксированных строках и колонках. Фиксированная строка **FixedRows** нам нужна только одна, это значение по умолчанию. А фиксированные столбцы **FixedCols** нам и вовсе не нужны, тут поставьте ноль.

Остальные параметры сетки можно оставить без изменений. Если вы всё сделали правильно, у вас в результате должна получиться такая форма:



**Рис. 16.1.** Главная форма проекта

Пойдем дальше. Прежде, чем приступать к самому кодированию, давайте сделаем еще одну форму - окно редактора контактов. Командой **Файл -> Создать форму** создайте новую форму. Присвойте ей имя **fEdit**, нажмите кнопку **"Сохранить все"** и дайте модулю формы имя **Edit**.

В свойстве **Caption** напишите "**Редактор контакта**". Почему не "контактов"? Каждый раз мы будем загружать в это окно по одному контакту, поэтому и использовано единственное число. В свойстве **BorderStyle** также выберите **bsDialog**, в свойстве **Position**- **poMainFormCenter**, чтобы окно появлялось по центру главной формы, где бы она ни находилась. Размеры формы: ширина **width = 400**, высота **height = 225** пикселей.

Теперь разместим на форме компоненты. Установите метку **TLabel**, в **Caption** которой напишите "**Укажите имя контакта:**".

Ниже установите строку **TEdit**. Переименуйте её в **eName**, ширину сделайте 380. Очистите свойство **Text**.

Ниже еще одну метку, с текстом в **Caption** "**Укажите телефон:**".

Ниже установите еще один **TEdit**. Переименуйте её в **eTelephone**, ширину сделайте

210. Очистите свойство **Text**.

Ниже еще одну метку, с текстом в **Caption** "**Выберите тип телефона:**".

Ниже нам потребуется список выбора **TComboBox**. Мы же не хотим, чтобы пользователю каждый раз самому приходилось писать "мобильный" или "служебный"? Переименуйте **TComboBox** в **CBNote**, откройте редактор свойства **Items** и там наберите следующие строки:

**Мобильный Домашний Служебный Соседский**

Если сможете придумать другие типы, можете дописать их сюда.

В свойстве **ItemIndex** компонента установите 0, чтобы в списке появилась первая строка "**Мобильный**". Ширину сделайте 210.

Ещё ниже установите на одном уровне кнопки **TBitBtn**. Первую переименуйте в **bSave**, в свойстве **Kind** выберите **bkOK**, чтобы на кнопке появилась картинка - зеленая галочка, убедитесь, что свойство **ModalResult** кнопки перешло в **mrOk**. В свойстве **Caption** напишите "**Сохранить контакт**". Чтобы текст нормально умещался, увеличьте её ширину до 160 пикселей.

Вторую кнопку переименуйте в **bCancel**, в свойстве **Kind** выберите **bkCancel**, а в **Caption** напишите "**Не сохранять**". Свойство **ModalResult**



кнопки должно перейти в **mrCancel**. В результате, у вас должна получиться вот такая форма:

**Рис. 16.2.** Форма проекта fEdit

Чтобы более не возвращаться к этой форме, давайте сразу введем весь необходимый код формы, тем более что его совсем мало. Для кнопок нам вообще не нужно писать код. Когда мы изменили их свойства **Kind**, установив там **bkOK** и **bkCancel**, сразу изменилось и их свойство **ModalResult**, установившись, соответственно, в **mrOK** и **mrCancel**. И теперь нажатие на любую из этих кнопок приведет к закрытию формы, и к тому, что в **ModalResult** формы попадет значение **ModalResult** нажатой кнопки. То есть, мы сможем узнать, какой кнопкой пользователь закрыл окно.

Но нам нужно все же сделать некоторую подготовительную работу. Прежде, чем показывать форму, нам нужно будет перевести фокус ввода в первый **TEdit**, ведь от прежней работы с формой выделенным мог остаться и другой компонент - кнопка или список выбора, например. А это будет неудобно для пользователя, ему придется лишний раз кликать мышью, выделяя первую строку.

Выделите форму, щелкнув по свободному участку, в Инспекторе объектов перейдите на вкладку **События** и сгенерируйте событие формы **OnShow**- это событие возникает



всякий раз перед показом формы - идеальное место для подготовительной работы. Код события совсем простой:

```
procedure TfEdit.FormShow(Sender: TObject); begin  
    eName.SetFocus; end;
```

И с редактором контактов это все! Вся остальная работа будет происходить в главной форме.

В редакторе форм перейдите на вкладку **Main** - наша главная форма. Прежде всего, в разделе **uses** главной формы, через запятую, добавьте модуль **edit** - это наш редактор контактов. Если мы не добавим его в раздел **uses**, то не сможем с ним работать из главной формы.

Далее, **перед** разделом глобальных переменных, тем, где объявлена переменная формы **fMain**, создадим еще один раздел **type**, в котором опишем запись:

```
type  
    Contacts = record Name:  
        string[100];  
        Telephon: string[20]; Note:  
        string[20];  
    end; //record
```

В этой записи три поля. Первое поле **Name** предназначено для имени контакта. Вы можете заметить, что оно ограничено размером 100 символов. В записи можно указывать неограниченный тип **string**, но тогда создать типизированный файл этого типа не получится. Поэтому у нас все строки имеют фиксированный размер. В поле **Name** пользователь будет вписывать имена контактов, например:

Анатолий Васильевич, директор Дима,  
друг  
Леночка, секретарь

Может показаться, что 100 символов для имени контакта - это перебор. Однако вспомните, что для текста у нас используется кодировка UTF-8, а в ней каждый символ кириллицы занимает 2 байта. То есть, если писать имена по-русски, реально можно вписать только 50

символов! А фамилии и имена бывают очень длинными, но 50 символов должно хватить. Для телефона мы указали 20 символов. Это обычно, цифры, скобки и знаки тире:

111-11-11

3-(222)-111-11-11

На каждый из этих символов требуется только 1 байт, так что 20 символов хватит за глаза для любого телефона.

Ну и, наконец, поле **Note**, в котором будет одно из значений:

**Мобильный Домашний Служебный Соседский**

Эти символы тоже занимают по 2 байта, самое длинное слово имеет 9 символов - 18 байт. А мы указали 20, так что должно хватить.

Далее, в разделе глобальных переменных, после описания самой формы, опишем глобальную переменную **adres**:

```
var
  fMain: TfMain;
  adres: string; //адрес, откуда запущена программа
```

Эта переменная понадобится нам позже, в нее мы запишем адрес, откуда была запущена программа, чтобы потом по этому адресу отыскать файл с контактами. Ранее мы говорили, что если не указывать адрес, то компьютер будет искать файл в текущей папке, то есть - в папке с программой. Однако так бывает не всегда. Если вы параллельно работаете с несколькими другими программами, и все они то и дело обращаются к файлам, то текущий адрес может измениться. Вот почему всегда лучше подстраховаться, и указать настоящий адрес - это гарантирует вашу программу от любых ошибок.

Далее, "научим" программу принимать новые данные и выводить их в сетку. Сгенерируйте событие **OnClick** для кнопки **"Добавить контакт"**. Код события такой:

```
procedure TfMain.bAddClick(Sender: TObject); begin
  //очищаем поля, если там что-то есть:
  fEdit.eName.Text:= ''; fEdit.eTelephone.Text:= '';
```

```

//устанавливаем ModalResult редактора           в mrNone:
fEdit.ModalResult:= mrNone;
//теперь выводим форму:
fEdit.ShowModal;
//если пользователь ничего не ввел - выходим:
if (fEdit.eName.Text= "") or (fEdit.eTelephone.Text= "") then exit;
//если пользователь не нажал "Сохранить" - выходим:
if fEdit.ModalResult <> mrOk then exit;
//иначе добавляем в сетку строку, и заполняем её:
SG.RowCount:= SG.RowCount + 1;
SG.Cells[0, SG.RowCount-1]:= fEdit.eName.Text; SG.Cells[1,
SG.RowCount-1]:= fEdit.eTelephone.Text; SG.Cells[2,
SG.RowCount-1]:= fEdit.CBNote.Text;
end;

```

Давайте разбираться с кодом. Прежде всего, в редакторе контактов **fEdit** мы очистили текст из строк с именем и номером телефона. Ведь если пользователь уже добавлял или редактировал контакты, в этих строках останутся записи с прошлого сеанса. Затем свойству **ModalResult** формы мы присваиваем значение **mrNone** - нет результата.

Делаем это затем, чтобы потом точно знать, каким образом пользователь закрыл окно редактора контактов. Если кнопкой **"Сохранить контакт"**, то в этом случае **ModalResult** формы будет **mrOk**, если кнопкой **"Не сохранять"**, **ModalResult** формы будет **mrCancel**. Если же он просто закрыл окно крестиком в правом верхнем углу окна или клавишами **<Alt+F4>**, то в этом случае **ModalResult** формы так и останется **mrNone**.

Далее, мы выводим окно редактора контактов, как модальное.

К следующей строке кода программа доберется, когда пользователь введет (или не введет) новый контакт и закроет окно редактора. Прежде, чем что-то добавлять в сетку, мы делаем проверку - а ввел ли пользователь имя нового контакта, и телефон? Ведь если он ничего не ввел, то и сохранять нечего. И если он ввел один только телефон, без имени, или имя без телефона - какой смысл сохранять такой контакт?

Поэтому, если нет имени или телефона, мы просто выходим из процедуры:

```

if (fEdit.eName.Text= "") or (fEdit.eTelephone.Text= "") then exit;

```

Однако может случиться, что пользователь ввел и имя, и телефон, но потом передумал сохранять контакт и нажал кнопку **"Не сохранять"**, или просто закрыл окно редактора. Поэтому нам нужно сделать еще одну проверку:

```
if fEdit.ModalResult <> mrOk then exit;
```

Здесь мы смотрим значение свойства **ModalResult** формы. Оно сможет стать **mrOK** только в одном случае - если пользователь нажал кнопку "**Сохранить контакт**". И если это не так, то выходим из процедуры, ничего не сохраняя.

Если мы не вышли из процедуры в предыдущих двух проверках, то это означает, что

1. нам есть, что сохранять
2. пользователь нажал "**Сохранить контакт**"

И если это так, значит, добавляем новую строку в сетку SG и заполняем три её колонки значениями из редактора контактов. Дополнительных проверок на существование текста примечания мы не делали, так как там в любом случае будет указан один из типов телефонов:

```
//иначе добавляем в сетку строку, и заполняем её:
SG.RowCount:= SG.RowCount + 1;
SG.Cells[0, SG.RowCount-1]:= fEdit.eName.Text; SG.Cells[1,
SG.RowCount-1]:= fEdit.eTelephone.Text; SG.Cells[2,
SG.RowCount-1]:= fEdit.CBNote.Text;
```

Сохраните проект и запустите его на выполнение. Если вы всё сделали верно, программа позволит вам добавлять новые контакты. Не обращайтесь внимания на недоработанный вид сетки - узкие столбцы без заголовков - это мы исправим чуть позже.

Но этого мало, хотелось бы ещё и сохранять эти контакты в файл. Для этого нам лучше всего подойдет событие формы **OnClose**, которое будет выполняться только один раз за сеанс работы программы - перед её закрытием. Форма вся закрыта компонентами, но по прошлым лекциям вы должны помнить, что форму можно выделить в верхнем окошке **Инспектора объектов**, где компоненты указаны в виде дерева, в самой вершине которого форма **fMain**. Еще можно выделить компонент верхнего над формой уровня - панель или сетку, и нажать **<Esc>**. При этом выделение перейдет на более низкий уровень - на форму.

Итак, выделите форму, в **Инспекторе объектов** перейдите на вкладку **События** и

сгенерируйте событие **OnClose**. Код

события такой:

```
procedure TfMain.FormClose(Sender: TObject; var CloseAction: TCloseAction); var
  MyCont: Contacts; //для очередной записи f: file of
  Contacts; //файл данных
  i: integer; //счетчик цикла begin
  //если строки данных пусты, просто выходим:
  if SG.RowCount = 1 then exit;
  //иначе открываем файл для записи:
  try
    AssignFile(f, adres + 'telephones.dat'); Rewrite(f);
    //теперь цикл - от первой до последней записи сетки: for i:= 1 to
    SG.RowCount-1 do begin
      //получаем данные текущей записи:
      MyCont.Name:= SG.Cells[0, i];
      MyCont.Telephon:= SG.Cells[1, i];
      MyCont.Note:= SG.Cells[2, i];
      //записываем их:
      Write(f, MyCont); end;
  finally CloseFile(f);
  end; end;
```

Как видите, мы объявили переменную типа нашей записи **Contacts**, и типизированный файл с таким же типом. Еще мы объявили целочисленную переменную, которая понадобится нам для организации цикла.

Вначале мы проверяем - есть ли в сетке данные? Если **SG.RowCount = 1**, то это означает, что в сетке есть только одна строка - фиксированная, та, где у нас находятся заголовки колонок (а меньше 1 строки и быть не может, пользователь не сможет удалить фиксированную строку). Другими словами, данных нет. В этом случае мы просто выходим из процедуры.

Если программа пошла дальше, значит, данные в сетке есть, правда? Тогда мы начинаем работу по сохранению данных в файл. Эту работу мы помещаем в блок **try- finally-end**, работу с файлами (в том числе типизированными) мы с вами неоднократно рассматривали, так что подробно останавливаться тут не буду. Но пару замечаний сделать все же нужно. Во-первых, при ассоциации с файлом мы указали не только имя, но и адрес файла, который должен находиться в переменной **adres**:

```
AssignFile(f, adres + 'telephones.dat');
```

Однако эту переменную мы объявили, но ничего туда пока не внесли. Не обращайтесь на это внимания - мы сделаем это в следующем коде. Пока считайте, что адрес там есть, однако не спешите запускать программу на выполнение, точнее, не пытайтесь закрыть эту программу, если там есть данные.

Во-вторых, запись данных в файл мы организовали в виде цикла:

```
for i:= 1 to SG.RowCount-1 do begin
```

То есть, мы делаем эту работу от строки с индексом 1 (нулевой индекс у фиксированной строки, она нам для обработки данных не нужна), до строки с индексом **SG.RowCount-1**. Так, если в сетке у нас 10 строк, то индекс последней строки = 9, а **RowCount** вернет 10, потому мы отнимаем единицу.

Внутри цикла мы считываем в переменную типа запись данные из сетки, из текущей строки данных:

```
MyCont.Name:= SG.Cells[0, i];
```

Когда считали все значения, мы записываем эту переменную в типизированный файл. Поскольку типы переменной и файла совпадают, то запись будет успешной, а указатель сместится в конец файла. Затем цикл продолжится. При каждом новом шаге цикла значение **i** будет увеличиваться на единицу, и код будет ссылаться на новую следующую строку. Записи будут записываться в файл одна за другой, до самого конца. В результате у нас получится нечто вроде базы данных.

Но чтобы от нашей базы был толк, программа при загрузке должна эти данные считывать обратно в сетку. Удобнее всего, как уже говорилось ранее, подготовительную работу помещать в событие формы **OnCreate**. Сгенерируйте это событие, код будет таким:

```
procedure TfMain.FormCreate(Sender: TObject); var
  MyCont: Contacts; //для очередной записи f: file of
  Contacts; //файл данных
  i: integer; //счетчик цикла begin
```

```

//сначала получим адрес программы:
adres:= ExtractFilePath(ParamStr(0));
//настроим сетку:
SG.Cells[0, 0]:= 'Имя';
SG.Cells[1, 0]:= 'Телефон';
SG.Cells[2, 0]:= 'Примечание';
SG.ColWidths[0]:= 365;
SG.ColWidths[1]:= 150;
SG.ColWidths[2]:= 150;
//если файла данных нет, просто выходим:
if not FileExists(adres + 'telephones.dat') then exit;
//иначе файл есть, открываем его для чтения и
//считываем данные в сетку:
try
  AssignFile(f, adres + 'telephones.dat'); Reset(f);
  //теперь цикл - от первой до последней записи сетки: while not
  Eof(f) do begin
    //считываем новую запись:
    Read(f, MyCont);
    //добавляем в сетку новую строку, и заполняем её:
    SG.RowCount:= SG.RowCount + 1;
    SG.Cells[0, SG.RowCount-1]:= MyCont.Name; SG.Cells[1,
    SG.RowCount-1]:= MyCont.Telephon;
  SG.Cells[2, SG.RowCount-1]:= MyCont.Note; end;
  finally CloseFile(f);
end; end;

```

Здесь мы, прежде всего, получаем адрес запускаемой программы в переменную **adres**:

```
adres:= ExtractFilePath(ParamStr(0));
```

В любом проекте доступен параметр **ParamStr(0)**. В этом параметре хранится адрес и имя загружаемой программы. Функция **ExtractFilePath** отсекает имя файла, возвращая только его адрес с завершающим "\", например:

**C:\Program Files\MyProg\**

Адрес, откуда запущена программа, мы и сохраняем в переменную **adres**. Поскольку она глобальная, то становится доступной во всем модуле главной формы. Поэтому мы воспользовались этой переменной для получения адреса в событии **OnClose**, в котором прописали код сохранения данных в файл. Так как событие **OnCreate** выполняется первым, то адрес в переменной **adres** действительно, уже будет. Так что теперь данные будут сохраняться.

Далее мы настраиваем сетку - указываем ширину столбцов, вписываем заголовки колонок. Как это делается, мы разбирали в прошлой лекции.

Затем мы смотрим - есть ли файл? Программа может запускаться впервые, или пользователь мог удалить старый файл, в любом из этих случаев мы выходим, не производя дальнейших действий. Сетка все равно уже настроена, так что программа будет выглядеть, как надо. Да и адрес программы мы уже поместили в переменную **adres**.

Но если файл есть, то нужно считать из него данные, и поместить их в сетку. Работа с файлом вам знакома, так что описывать процесс не буду. Замечу только, что в данном случае мы воспользовались циклом

**while not Eof(f) do begin**

То есть, делать, пока не конец файла. При открытии файла его указатель устанавливается в начало. Когда мы считываем первую запись в переменную **MyCont**, то указатель перемещается к следующей записи. Так, раз за разом, считывая запись в переменную, а затем, копируя из нее данные в сетку, мы движемся до конца файла, обрабатывая все записи, какие там есть. Причем при каждом шаге цикла мы сначала добавляем в сетку новую строку:

**SG.RowCount:= SG.RowCount + 1;**

а затем в эту, последнюю строку, мы записываем считанные из файла данные.

Итак, на данном этапе, мы научили программу добавлять новые записи (событие **OnClick** кнопки **bAdd**), научили её сохранять эти записи в файл (событие формы **OnClose**) и считывать их из файла (событие формы **OnCreate**). Осталось научить программу редактировать существующие записи - вдруг пользователь что то решит исправить, например, сменился телефон контакта, удалять контакты, ставшие ненужными, а также сортировать список по алфавиту. Ведь пользователь будет вносить контакты, как придется, не по порядку, а нужную запись гораздо проще найти в сортированном списке.

Итак, сгенерируйте событие **OnClick** для кнопки **bEdit** ("Редактировать контакт"). Код:

**procedure TfMain.bEditClick(Sender: TObject); begin**



```

//если данных в сетке нет - просто          ВЫХОДИМ:
if SG.RowCount = 1 then exit;
//иначе записываем данные в форму редактора:
fEdit.eName.Text:= SG.Cells[0, SG.Row];
fEdit.eTelephone.Text:= SG.Cells[1, SG.Row];
fEdit.CBNote.Text:= SG.Cells[2, SG.Row];
//устанавливаем ModalResult редактора в mrNone:
fEdit.ModalResult:= mrNone;
//теперь выводим форму:
fEdit.ShowModal;
//сохраняем в сетку возможные изменения,
//если пользователь нажал "Сохранить":
if fEdit.ModalResult = mrOk then begin SG.Cells[0, SG.Row]:=
    fEdit.eName.Text; SG.Cells[1, SG.Row]:=
    fEdit.eTelephone.Text; SG.Cells[2, SG.Row]:=
    fEdit.CBNote.Text;
end; end;

```

Давайте разбираться с кодом. Как обычно, мы выходим из процедуры, если данных в сетке нет - редактировать ведь нечего. Иначе перед показом окна редактора контактов мы прodelываем подготовительную работу - заносим в строки **TEdit** этой формы данные контакта - имя и телефон. Также свойству **Text** списка выбора **TComboBox** мы присваиваем текст примечания. Все это позволит представить пользователю данные текущей записи. Вы помните, как узнать индекс текущей строки сетки? Этот индекс храниться в свойстве **Row**:

```
fEdit.eName.Text:= SG.Cells[0, SG.Row];
```

Далее, **ModalResult** формы редактора мы устанавливаем в **mrNone**, чтобы потом можно было определить, каким образом пользователь закрыл этот редактор.

Потом мы выводим форму, как модальное окно. В заключение мы проверяем - нажал ли пользователь кнопку **"Сохранить контакт"**? Если да, то мы перезаписываем данные в сетке на новый вариант, из формы редактора.

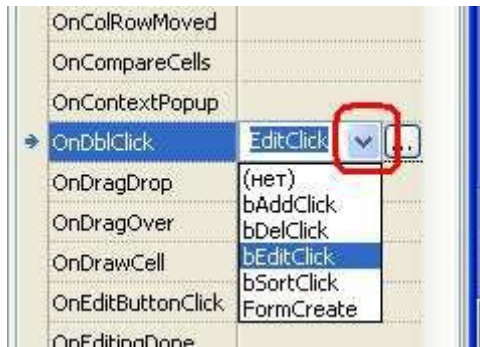
Нажав на кнопку **"Редактировать контакт"**, пользователь сможет воспользоваться этим кодом. Однако в хороших программах пользователю часто предоставляют различные инструменты для выполнения одной задачи. Например, хорошим тоном считается редактировать запись сетки при двойном клике по ней. Давайте, реализуем эту

возможность, тем более что это совсем

просто. Выделите сетку. В

## Инспекторе

**объектов** перейдите на вкладку **События**. Только не торопитесь генерировать новое событие, ведь писать новый код нам не придется. В событии **OnDblClick** нажмите кнопку



выбора, и выберите там событие **bEditClick** - то есть то, которое мы только что описали:

**Рис. 16.3.** Выбор уже существующего события

Таким образом, и при нажатии кнопки **"Редактировать контакт"**, и при двойном клике по записи сетки будет выполнена одна и та же процедура.

Теперь научим программу удалять контакты. Сгенерируйте **OnClick** для кнопки **"Удалить контакт"**:

```
procedure TfMain.bDelClick(Sender: TObject); begin
  //если данных нет - выходим:
  if SG.RowCount = 1 then exit;
  //иначе выводим запрос на подтверждение:
  if MessageDlg('Требуется подтверждение',
    'Вы действительно хотите удалить контакт "' + SG.Cells[0,
    SG.Row] + '"?',
    mtConfirmation, [mbYes, mbNo, mbIgnore], 0) = mrYes then
    SG.DeleteRow(SG.Row);
end;
```

Этот код проще, да? Опять мы выходим из процедуры, если данных нет, ведь тогда и удалять нечего. Удалить запись просто, но нужно ли это делать? Пользователь мог нажать на кнопку не подумав, или случайно. Поэтому перед удалением мы запросим подтверждения:

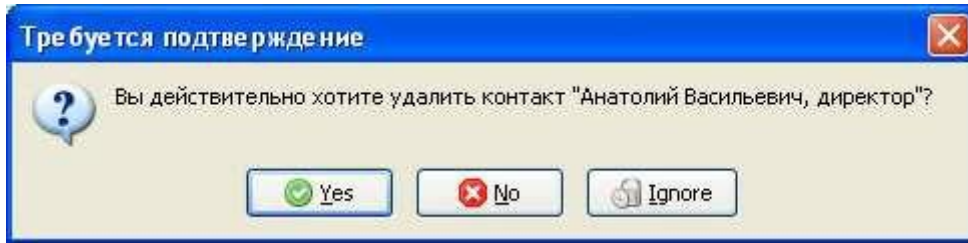
```
if MessageDlg('Требуется подтверждение',
```

```

'Вы действительно                               хотите удалить контакт
'' + SG.Cells[0, SG.Row] + ''?',
mtConfirmation, [mbYes, mbNo, mbIgnore], 0) = mrYes then
  SG.DeleteRow(SG.Row);

```

Программа выведет подобное сообщение:



**Рис. 16.4.** Запрос на подтверждение

И только если пользователь подтвердит удаление контакта, он будет удален:

```
SG.DeleteRow(SG.Row);
```

Нам осталось сгенерировать событие **OnClick** для кнопки **"Сортировать список"**. Его код:

```

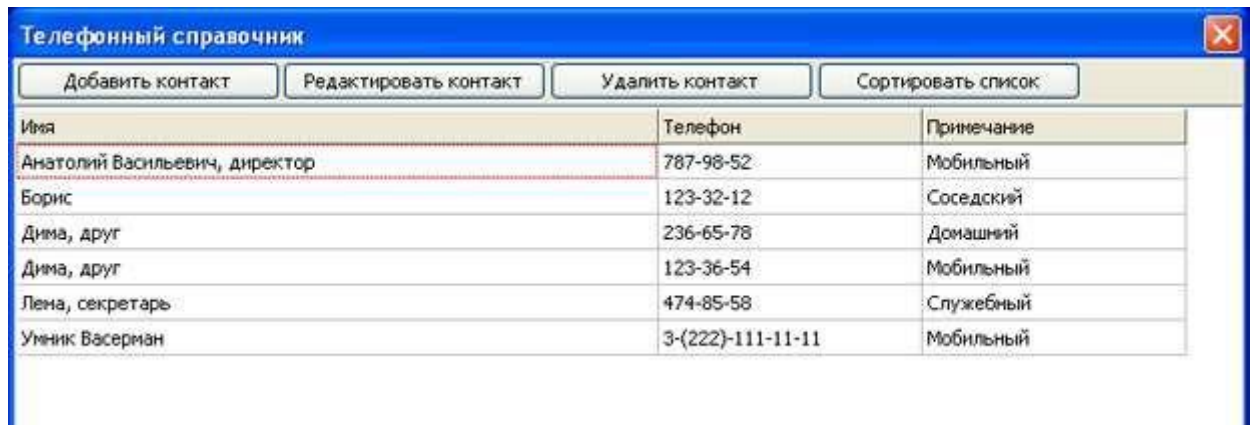
procedure TfMain.bSortClick(Sender: TObject); begin
  //если данных в сетке нет - просто выходим:
  if SG.RowCount = 1 then exit;
  //иначе сортируем список:
  SG.SortColRow(true, 0); end;

```

Код достаточно простой, мы выполняем сортировку, если есть данные, по первому столбцу (с именами контактов):

```
SG.SortColRow(true, 0);
```

Сохраните проект, запустите его и попробуйте в работе. Приложение должно работать, и выглядеть примерно так:



**Рис. 16.5.** Окно работающего приложения

Если есть желание, можете дополнительно расширить проект - самостоятельно сделать окно "**О программе**" к этому приложению, реализовать механизм сортировок иначе, где пользователь мог бы выбрать способ сортировки: по именам контактов, по телефонам, или по типам телефонов.

После автономной отладки включите приложение как модуль с вызовом из главного меню в свой учебный проект.

## 15. Практическое занятие 8. Синтез внутри машинного информационного обеспечения сложной системы

CA ERwin Data Modeler (далее ERwin) - CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных.

Работа с программой начинается с создания новой модели, для которой нужно указать тип и целевую СУБД (рис.1).

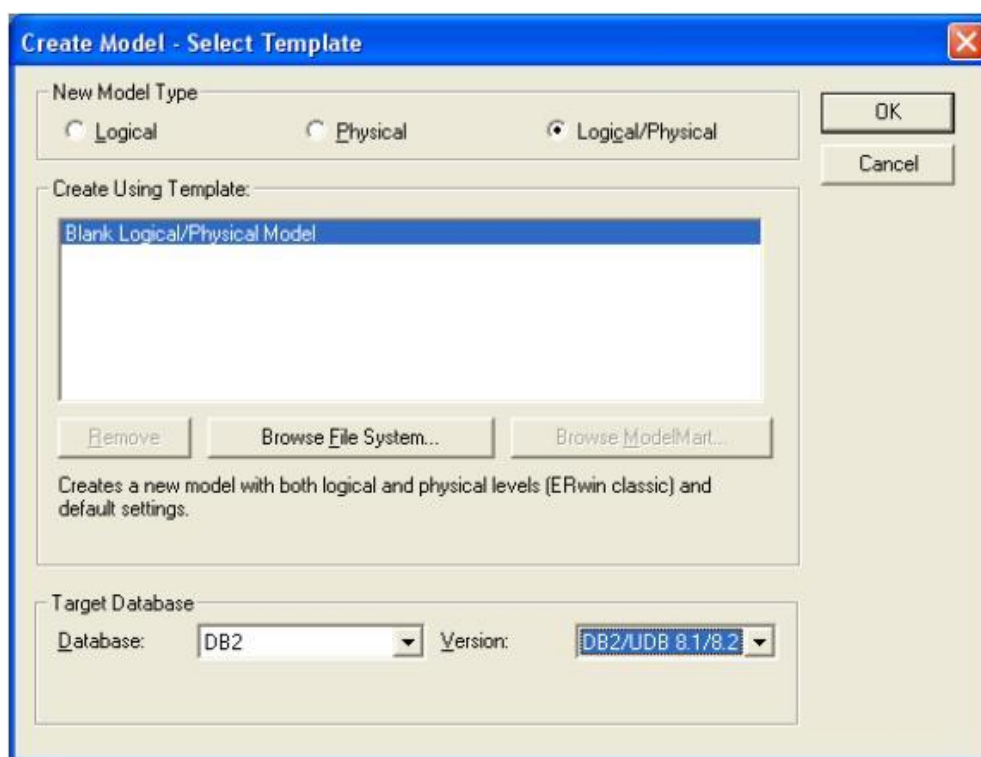


Рисунок 1. Создание новой модели

ERwin позволяет создавать логическую, физическую модели и модель, совмещающую логический и физический уровни.

**Логический уровень** - это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире (например "Постоянный клиент", "Отдел" или "Заказ").

Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. Логическая модель данных является универсальной и никак не связана с конкретной реализацией СУБД.

**Физический уровень** зависит от конкретной СУБД. В физической модели содержится информация о всех объектах БД. Физическая модель зависит от конкретной реализации СУБД. Одной и той же логической модели могут соответствовать несколько разных физических моделей.

На логическом уровне ERwin поддерживает две нотации (IE и IDEF1X), на физическом - три (IE, IDEF1X и DM). Далее будет рассматриваться работа с ERwin в нотации IDEF1X.

Переключение между логической и физической моделями данных осуществляется через список выбора на стандартной панели (рис.2).

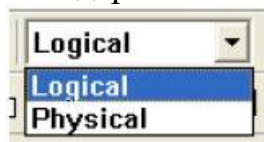


Рисунок 2.Переключение между уровнями

**Примечание.** В созданной модели с настройками по умолчанию некорректно отображаются русские символы. Чтобы устранить этот недостаток, необходимо подкорректировать используемые в модели шрифты. Для этого необходимо зайти в меню Format -> Default Fonts & Colors, последовательно пройти по всем вкладкам, в качестве шрифта выбрав любой шрифт, название которого заканчивается на CYR (например, Arial CYR), и выставив переключатель Apply To в значение All Objects.

### Логический уровень модели данных

Для создания на логическом уровне сущностей и связей между ними предназначена панель *Toolbox*:

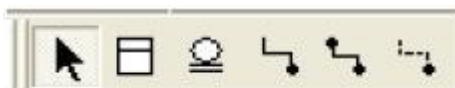







Рисунок 3.Панель Toolbox

| Вид кнопки   | Назначение кнопки   |
|--|---|
|  | Создание новой сущности. Для этого нужно щелкнуть по кнопке и затем по свободному месту на модели   |
|  | Создание категории. Для установки категориальной связи нужно щелкнуть по кнопке, далее - по сущности-родителю, и затем - по сущности-потомку.       |
|  | Создание идентифицирующей связи. Для связывания двух сущностей нужно щелкнуть по кнопке, далее - по сущности-родителю, затем - по сущности-потомку. |
|  | Создание связи "многие ко многим"   |
|  | Создание неидентифицирующей связи   |

После создания сущности ей нужно задать атрибуты. Для этого нужно дважды щелкнуть по ней или в контекстном меню выбрать пункт *Attributes* (рис.4).

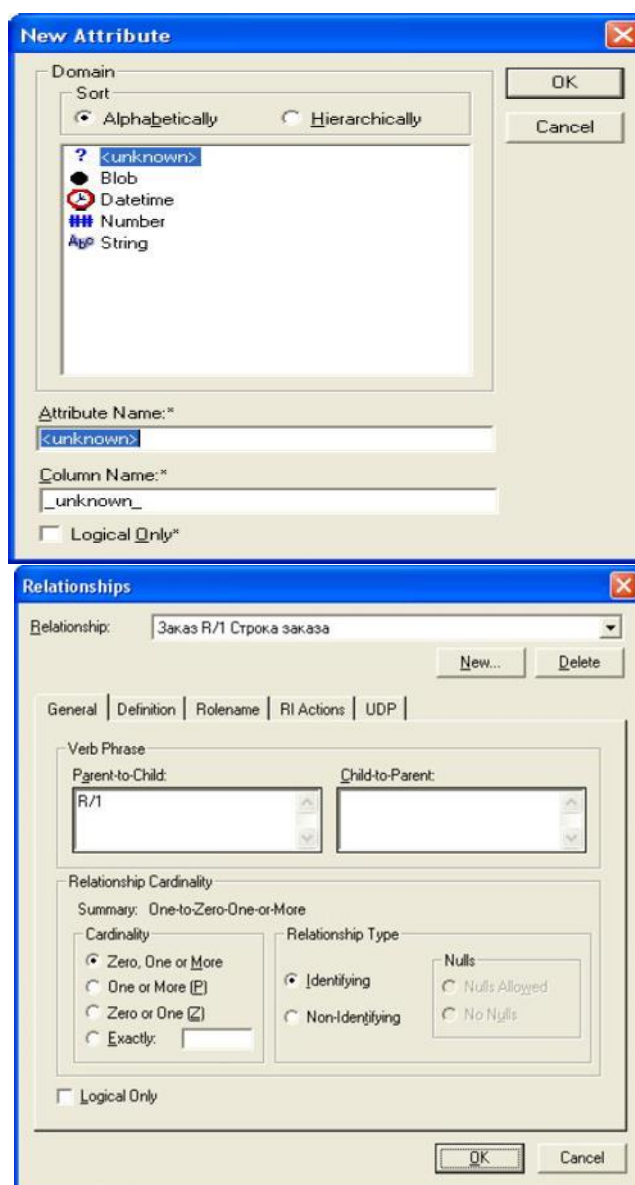


Рисунок 4.Окно атрибутов выбранной сущности

В появившемся окне можно просмотреть и отредактировать информацию о созданных атрибутах, создать новые. Здесь же задается первичный ключ. Для создания нового атрибута следует нажать кнопку *New*. В появившемся окне можно выбрать тип атрибута (BLOB, дата/время, число, строка), задать имя атрибута (*Attribute Name*) и имя столбца (*Column Name*), который будет соответствовать атрибуту на физическом уровне (рис.5).

Рисунок 5.Окно создания атрибута

После создания сущностей создаются связи между ними. При создании идентифицирующей связи атрибуты, составляющие первичный ключ сущности-родителя, мигрируют в состав первичного ключа сущности-потомка, при создании неидентифицирующей связи - просто в состав атрибутов сущности-потомка. Задать свойства связи или поменять ее тип можно дважды щелкнув по ней или выбрав в контекстном меню пункт *Relationship Properties* (рис. 6). Здесь во вкладке *General* можно задать имя связи (в направлении родитель-потомок и потомок-родитель), мощность связи (ноль, один или больше; один и больше (P); ноль или один (Z); точно (конкретное число) ), поменять тип связи. Во вкладке *RI Action* можно задать ограничения целостности.



Пример логической модели базы

данных приведен на рис. 7.

Рисунок 6. Окно свойств связи

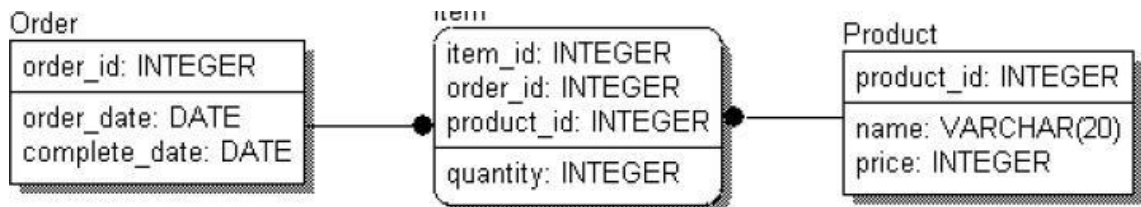


Рисунок 7. Пример логической схемы БД

Физический уровень модели данных

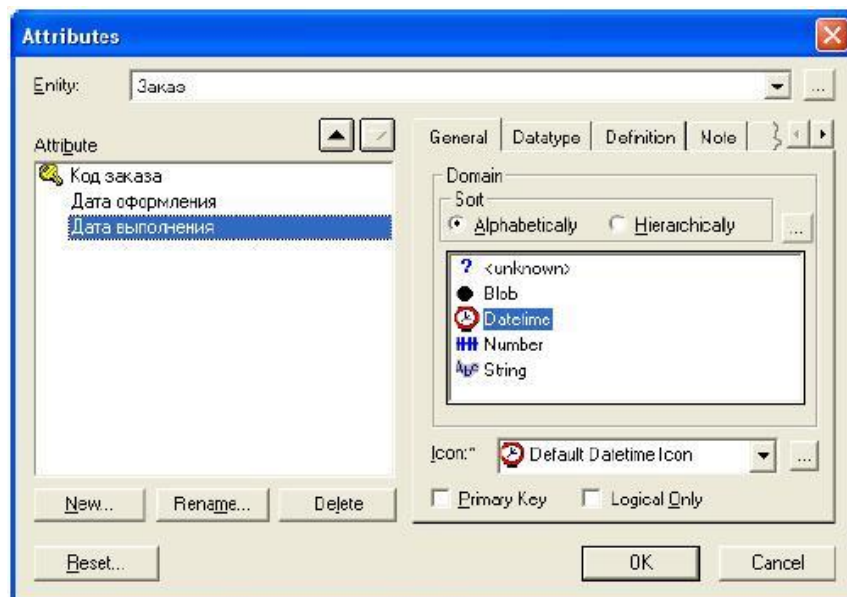
При переключении с логического уровня на физический автоматически будет создана физическая схема базы данных (рис.8)



Рисунок 8. Автоматически созданная физическая схема БД

Ее можно дополнить, отредактировать или изменить. Принципы работы с физической схемой аналогичны принципам работы с логической схемой.

По готовой физической схеме можно сгенерировать скрипты для выбранной СУБД. Для этого



предназначен пункт меню *Tools -> Forward Engineering/Schema Generation* (рис.9).



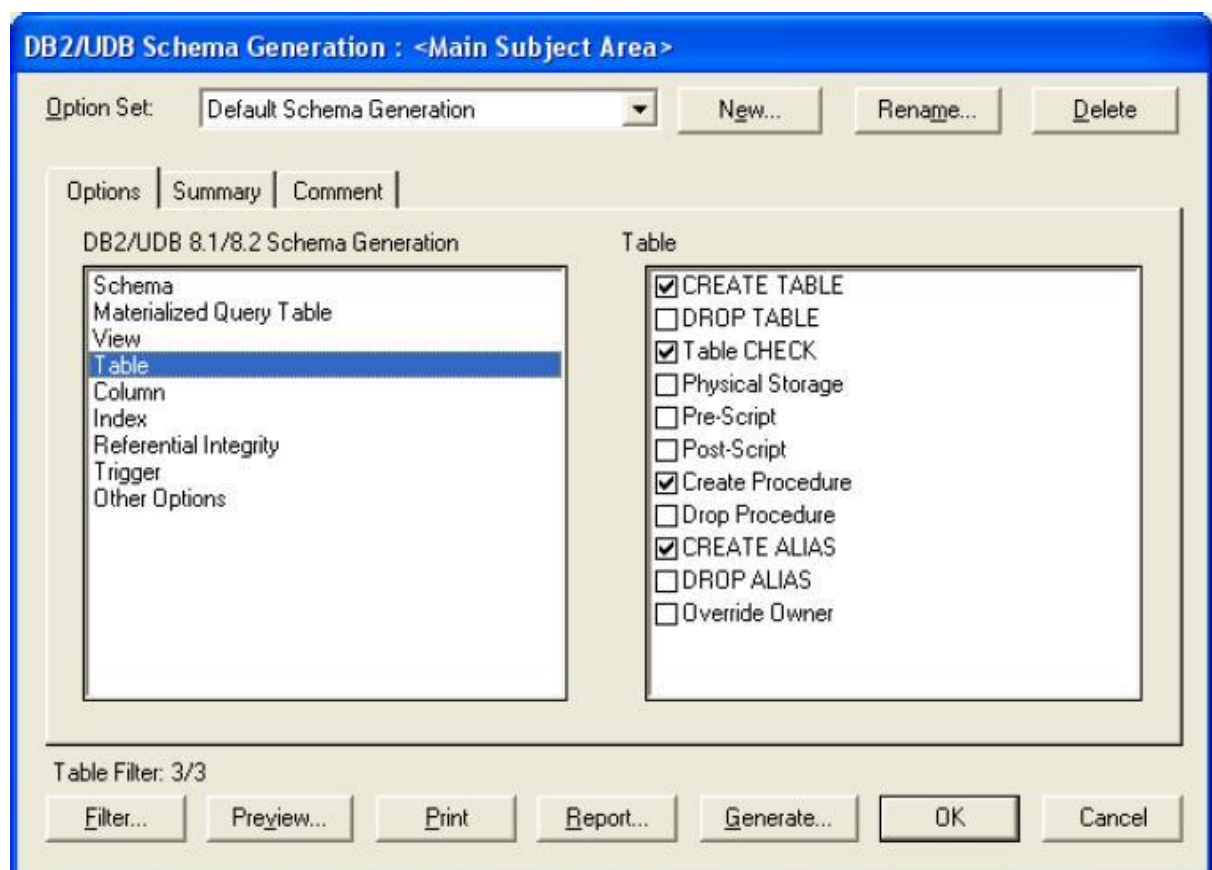


Рисунок 9. Окно генерации SQL-скриптов для целевой СУБД\_\_

Здесь можно указать, какие именно скрипты следует генерировать, предварительно просмотреть их и непосредственно сгенерировать (при этом ERwin произведет подключение к целевой СУБД и в автоматическом режиме выполнит все SQL-скрипты).

## 16. Лабораторная работа 8. Построение реляционной схемы данных.

### Цель работы:

построить реляционную схему данных выбранной предметной области в нотации IDEF1X. В данной лабораторной работе необходимо построить в нотации IDEF1X в CASE-средстве ERwin Data Modeler логическую схему данных предметной области, бизнес-процессы которой моделировались в предыдущих лабораторных работах.

**Примечание.** При построении модели можно ограничиться 5-6 сущностями.

### IDEF1X

IDEF1X основан на подходе Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме. Нотация Чена и сам процесс построения диаграмм сущность-связь изучалась в курсе "Организация баз данных и знаний", поэтому здесь мы рассмотрим только отличия IDEF1X от нотации Чена.

*Сущность (Entity)* - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности.

*Атрибут (Attribute)* - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Наименование атрибута должно быть выражено существительным в единственном числе.

*Связь (Relationship)* - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области.

В методе IDEF1X все сущности делятся на зависимые и независимые от идентификаторов.

Сущность является независимой от идентификаторов или просто независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. Сущность называется зависимой от идентификаторов или просто зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности. Независимая сущность изображается в виде обычного прямоугольника, зависимая - в виде прямоугольника с закругленными углами.

В IDEF1X существуют следующие виды мощностей связей:

**N** мощность - каждый экземпляр сущности-родителя может иметь ноль, один или более одного связанного с ним экземпляра сущности-потомка (по умолчанию);

**R** мощность - каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;

**Z** мощность - каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;

**конкретное число** - каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

Связь изображается линией, проводимой между сущностью-родителем и сущностью-потомком, с точкой на конце линии у сущности-потомка. По умолчанию мощность связи принимается равной N. Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется идентифицирующей, в противном случае — неидентифицирующей. Идентифицирующая связь изображается сплошной линией,

неидентифицирующая - пунктирной линией.

В ERwin'e при установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Эта операция называется миграцией атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ (FK). При установке неидентифицирующей связи атрибуты первичного ключа родительской сущности мигрируют в состав неключевых полей дочерней сущности.

**Построение логической модели данных предприятия по сборке и продаже компьютеров и ноутбуков.** Построение модели данных начинается с выделения сущностей данной предметной области. В нашем случае были выделены следующие сущности:

клиент - человек, который покупает компьютеры

заказ - список компьютеров, которые покупает клиент

компьютер

комплектующие - то, из чего собирают компьютеры

сотрудник - сотрудник предприятия, собирающий конкретный компьютер

Далее рассмотрим связи между сущностями:

Клиент - Заказ. Один клиент может делать несколько заказов. При этом если данные о клиенте имеются в базе данных, то он сделал минимум один заказ. Поэтому мощность связи - Р. Связь идентифицирующая, т.к. заказ без клиента существовать не может;

Заказ - Компьютер. В рамках одного заказа клиент может заказать несколько компьютеров, но как минимум заказ должен состоять из одного компьютера. Поэтому мощность связи - Р. Связь идентифицирующая, т.к. компьютер без заказа существовать не может;

Компьютер - Комплектующие. В состав одного компьютера входит много различных комплектующих; один и тот же тип комплектующего может входить в состав разных компьютеров. Мощность связи - много ко многим. В IDEF1X такой тип связи отсутствует, поэтому вводим промежуточную (ассоциативную) сущность - Конфигурация. Мощность связи между сущностями Компьютер и Конфигурация - Р, поскольку у любого компьютера должна быть конфигурация, мощность между сущностями Комплектующие и Конфигурация - N, поскольку какие-то комплектующие еще могут быть не установлены ни в один компьютер. Связь в обоих случаях идентифицирующая, т.к. конфигурация компьютера не может существовать без привязки к самому компьютеру и к комплектующим;

Комплектующие - Тип комплектующих. Поскольку перечень типов комплектующих, которые могут быть установлены в компьютер, ограничен, но используется очень часто, то мы приняли решение создать еще одну сущность - Тип комплектующих. Мощность связи - Р. Связь идентифицирующая;

Компьютер - Сотрудник. Каждый компьютер собирается каким-то одним сотрудником. Какие-то сотрудники могут собирать множество компьютеров. Мощность связи - N. Тип связи - неидентифицирующая, поскольку экземпляр сущности Компьютер уже может существовать, но за ним еще может быть не закреплен ни один сотрудник. Именно из этих же соображений в свойствах этой связи мы выбрали переключатель "Nulls Allowed" (на диаграмме это отображается в виде незакрашенного ромбика со стороны сущности-родителя.

Итоговая диаграмма показана на рис. 1:



Рисунок 1. Логическая модель данных предприятия по сборке компьютеров и ноутбуков

#### Содержание отчета:

логическая модель данных предметной области

краткое описание каждой сущности