МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
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

Методические указания к практическим занятиям 1-4

по дисциплине

Модуль 2. Информационные технологии и программирование (Основы алгоритмизации и программирования)

Ростов-на-Дону

2021

Методические указания к практическим занятиям 1– 4

по дисциплине
Модуль 2. Информационные технологии и программирование (Основы алгоритмизации и программирования)

Для студентов очной и заочной форм обучения
Направление подготовки - **09.03.01 «**Информатика и вычислительная техника»

Составитель: П.В. Лобзенко, доцент кафедры ИВТ
Рассмотрено и одобрено
на заседании кафедры ИВТ
Протокол от «30» августа 2021 г. № 1

Методические указания к практическому занятию 1

Технологии алгоритмизации типовых вычислительных задач

1. Цели занятия:

Практическое освоение составления блок-схем алгоритмов вычислительных задач. Практическое применение правил и приемов составления алгоритмов типовых вычислительных задач. Практическое изучение требований к составлению блок-схем алгоритмов.

2. Рекомендации:

Изучить справочный материал, приведенный ниже.

Структурное и неструктурное программирование

Различают три вида вычислительного процесса, реализуемого программами: линейный, разветвленный и циклический.

Линейная структура процесса вычислений предполагает, что для получения результата необходимо выполнить некоторые операции в определенной последовательности.

Разветвленная структура процесса вычислений предполагает, что конкретная последовательность операций зависит от значений одной или нескольких переменных.

Циклическая структура процесса вычислений предполагает, что для получения результата некоторые действия необходимо выполнить несколько раз.

Алгоритм – система точных и понятных предписаний о содержании и последовательности выполнения конечного числа действий, необходимых для решения любой задачи данного типа (класса).

Понятие возникло и используется давно. Сам термин «алгоритм» ведёт начало от перевода на европейские языки имени арабского математика Аль-Хорезми (IX век). Им были описаны правила (в нашем понимании - алгоритмы) выполнения основных арифметических действий в десятичной системе счисления.

Задача составления алгоритма не имеет смысла, если не известны, или не учитываются возможности его исполнителя (ребёнок может прочесть, но не может решить сложную задачу).

Исполнителем может быть не только человек, но и автомат. Компьютер – лишь частный, но наиболее впечатляющий пример исполнителя, чьё поведение основано на реализации алгоритма. Более того, создание персонального компьютера оказало воздействие на развитие теории алгоритмов, одной из областей дискретной математики.

Эффективный метод построения алгоритма – метод пошаговой детализации (последовательного построения). При этом сложная задача разбивается на ряд более простых. Для каждой подзадачи – свой алгоритм. Универсальный эффективный метод построения алгоритма является основой структурного программирования (языки QBasic, Turbo Pascal и др.).

Если алгоритм разработан, то его можно вручить разным людям (пусть и не знакомым с сутью решаемой задачи) и они, следуя системе правил, будут действовать одинаково и получат (при безошибочных действиях) одинаковый результат.

Используются различные способы записи алгоритмов:

- словесный (запись рецептов в кулинарной книге, инструкции по использованию технических устройств…);

- графический ;

- структурно-стилизованный (для записи используется язык псевдокода);

Свойства алгоритма.

При составлении и записи алгоритма необходимо обеспечить, чтобы он обладал рядом свойств:

Однозначность алгоритма – единственность толкования исполнителем правил выполнения действий и порядка их выполнения. Чтобы алгоритм обладал этим свойством он должен быть записан командами из системы команд исполнителя (сложить А и В).

Конечность алгоритма – обязательность завершения каждого из действий, составляющих алгоритм, и завершимость алгоритма в целом. Записанный на рис. 1 алгоритм обладает этим свойством.

Результативность алгоритма – предполагает, что выполнение алгоритма должно завершиться получением определённых результатов. У нас для целых А и В всегда будет вычислена сумма.

Массовость – возможность применения данного алгоритма для решения целого класса задач, отвечающих общей постановке задачи. В нашем Примере алгоритмом используется обозначение, а не конкретные числа, поэтому он может быть использован для сложения любых целых чисел.

Правильность алгоритма – способность алгоритма давать правильные результаты решения поставленных задач.

Способы задания алгоритма:

• словесный, (недостаток–многословность, возможна неоднозначность–«он встретил ее на поле с цветами»),

• табличный (физика, химия и т. д.),

• графический (блок-схемы).

Обозначения, применяемые в блок-схемах алгоритмов, приведены в таблице 1.1.

Таблица 1.1 – Обозначения, используемые в блок-схемах алгоритмов

Название блока	Обозначение	Назначение блока
Терминатор	Действие	Начало, завершение программы или подпрограммы
Процесс	Действие	Обработка данных (вычисления, пересылки и т. п.)
Данные	Данные	Операции ввода-вывода
Решение	Условие	Ветвления, выбор, итерационные и поисковые циклы
Подготовка	Действия	Счетные циклы
Граница цикла	Начало Конец	Любые циклы
Предопределенный процесс	Имя	Вызов процедур
Соединитель	Имя	Маркировка разрывов линий
Комментарий	Комментарий	Пояснения к операциям

Примеры базовых структур алгоритмов приведены на рисунке 1.1.

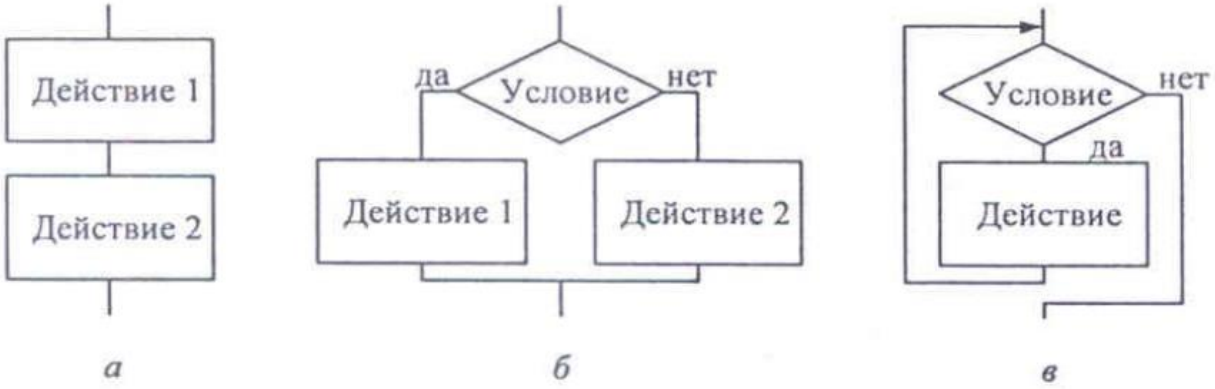


Рисунок 1.1- Базовые алгоритмические структуры:

а- следование; б- ветвление; в- цикл «пока».

Пример разработки схемы разветвляющегося алгоритма

Постановка задачи. Разработать схему алгоритма вычисления функции

Уяснение задачи. Анализ выражения для вычисления значения Y показывает, что в схеме алгоритма должна быть предусмотрена проверка значения X, после чего вычисляется Y согласно одному из двух заданных выражений. Схема алгоритма представлена на рис. 1.2.

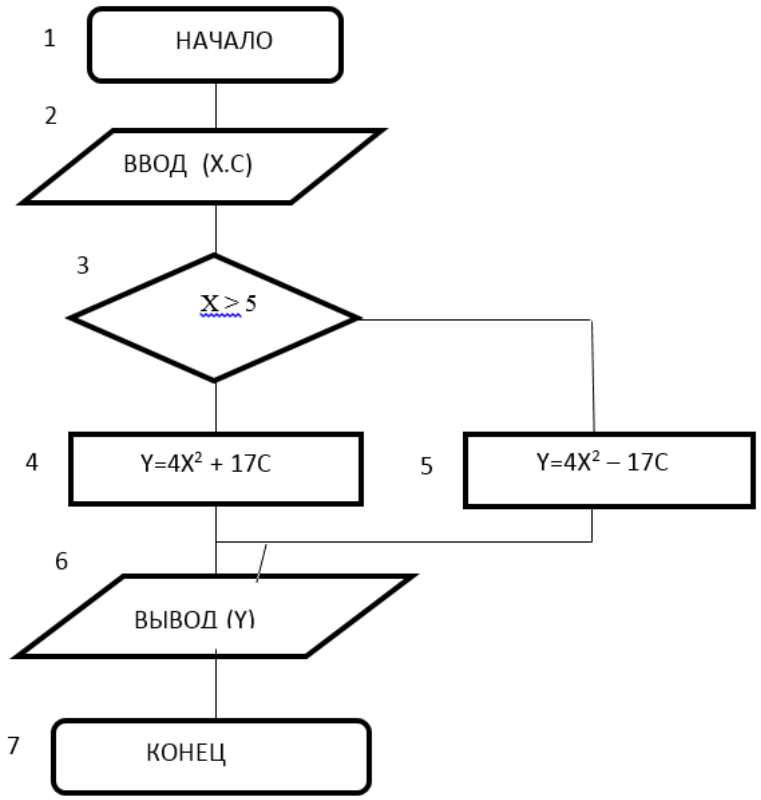


Рисунок 1.2- Пример разветвляющегося алгоритма

3. Порядок выполнения задания:

3.1. Выбрать первое (очередное) задание из №№1-4.

3.2. Выбрать первый (очередной) вариант из задания.

3.3. Прочитать и уяснить задачу.

3.4. Выполнить задачу: составить алгоритм задачи или по составленному алгоритму записать исходное выражение (сформулировать задачу).

3.5. Выполнить по очереди все варианты всех заданий.

3.6. Представить отчет в электронном виде на проверку.

4. Варианты заданий:

Составить алгоритмы и нарисовать блок-схемы следующих задач.

Варианты задания № 1	**Варианты задания № 2**
1. $Y = (A + \frac{B}{C}) \cdot 4D;$	1. $F = 1 - e^{4\cos X};$

2. $Y = (X_1 + \dfrac{A_1}{X^2 + \dfrac{A_2}{X_3 + A_3}};$	2. $F = \dfrac{\sin(XY - e^x)^2}{1 + \dfrac{X}{Y} 2{,}05 + 0{,}01 e^{X^2}};$
3. $Y = \dfrac{2AB}{P} \cdot \dfrac{A^2 + B^2}{(X^2 - B^2 - A^2)^2 + 4AX^3};$	3. $F = \sqrt{A^2 - X^2} \ln A + \dfrac{\lg A}{\sqrt{A^2 - X^2} + 1};$
4. $Y = \dfrac{3{,}84X + 10^4}{0{,}683X + 10^4};$	4. $F = \dfrac{1}{M\sqrt{AB}} arctg(e^{MX} \sqrt{\dfrac{A}{B}});$
5. $Y = \dfrac{A}{B + C} + \dfrac{D}{C^2};$	5. $F = e^{\sqrt{X}} \sqrt{\sqrt{X} + \sqrt{X^2 + A^2}};$
6. $Y = (\dfrac{1 + X}{1 - X})^2 + K^2 X^3;$	6. $F = \sqrt{X^2 + A^2} arctg \dfrac{X}{A} - \dfrac{\ln B}{tg \dfrac{X}{A}};$
7. $Y = 1 + \dfrac{2}{3} T^2 - \dfrac{2}{5} T^4;$	7. $F = \sqrt{X^2 + A^2} \dfrac{X}{A} - \dfrac{\lvert \lg B \rvert}{tg \dfrac{X}{A}};$
8. $Y(A + B)(A - B)^3$	8. $F = \dfrac{1}{M\sqrt{AB}} tg(e^{MX} \sqrt{\dfrac{A}{B}});$
9. $Y = 2^{K-1};$	

Варианты задания №3

1. Пусть даны длины сторон треугольника. Вычислите его площадь.

2. Вычислите расстояние между двумя точками на плоскости с данными координатами (x1,y1) и (x2,y2).

3. Введите положительное число **a.** Вычислите:

- площадь равностороннего треугольника со стороной **a;**
- площадь квадрата со стороной **a;**
- площадь круга с диаметром **a.**

4. Вычислите длину окружности, площадь круга, объем шара заданного радиуса.

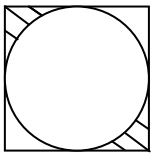
5. Пусть даны числа a, b, c. Найти площадь треугольника, две стороны которого равны a, b, а угол между этими сторонами равен c. Считать, что c – это: радианная мера; градусная мера.

6. Пусть известны длины сторон a,b,c треугольника. Вычислите высоты этого треугольника по формулам: $h_a = \frac{2}{a}\sqrt{p(p-a)(p-b)(p-c)}$ где $p = \frac{a+b+c}{2}$

$h_b = \frac{2}{b}\sqrt{p(p-a)(p-b)(p-c)}$ $h_c = \frac{2}{c}\sqrt{p(p-a)(p-b)(p-c)}$

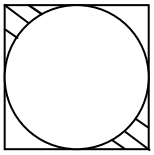
7. Даны координаты вершин некоторого треугольника. Вычислить его периметр.

8. В квадрат вписана окружность. Определить площадь заштрихованной части фигуры, если известна длина сторон квадрата.



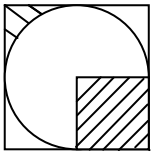
9. В квадрат вписана окружность.

Определить площадь заштрихованной части фигуры, если известен радиус окружности.



10. В квадрат вписана окружность. Определить площадь

заштрихованной части фигуры, если известна длина стороны квадрата.



Варианты задания №4

Разработать алгоритм вычисления функции. Значения аргументов установить самостоятельно.

1. $F(X)=\begin{cases}1, & при\ \ X<1;\\ 2X, & при\ \ 1\le X<2;\\ 4X^2, & при\ \ X\ge 2.\end{cases}$

2. $Y=\begin{cases}AX^2+BX+C, & если\ \ X\ge -25{,}5.\\ 25{,}5+X, & если\ \ X\ge -25{,}5\end{cases}$

3. $F=\begin{cases}A^2+B, & при\ \ A\le B;\\ A+B^3, & при\ \ A>B.\end{cases}$

4. $F=\begin{cases}X^2+4, & при\ \ X^2+AX+B<0;\\ X, & при\ \ X^2+AX+B\ge 0.\end{cases}$

5. $Y=\begin{cases}\dfrac{A}{A+B}, & при\ \ X+B>A^2;\\ 1+X^3, & при\ \ X+B\le A^{2.}\end{cases}$

6. $Y=\begin{cases}A^2+B^2+C^2, & если\ \ |A|\ge B+C;\\ \dfrac{1}{1+A+B+C}, & если\ \ |A|<B+C.\end{cases}$

7. $Y=\begin{cases}A^2+B, & \text{если}\ \ \text{A}<\text{B};\\ AB, & \text{если}\ \ \text{A}\ge B.\end{cases}$

8. $Y=\begin{cases}X, & \text{при}\ \ \text{X}^2+5X+21>100;\\ 0, & \text{при}\ \ \text{X}^2+5X+21\le 100.\end{cases}$

9. $Y=\begin{cases}0, & \text{если}\ \ \text{X}<\text{A}^2+2B;\\ AX+B^2, & \text{если}\ \ \text{X}\ge\text{A}^2+2B.\end{cases}$

10. $Y=\begin{cases}X^2+4, & \text{если}\ \text{X}^2+AX+B^2<0;\\ X^3, & \text{если}\ \ \text{X}^2+AX+B^2\ge 0.\end{cases}$

Методические указания к практическому занятию 2

Управляющие конструкции языка ТП 7.0

Цели занятия: Практически освоить основные управляющие операторы языка. Практически потренироваться в их использовании при составлении программ.

2. Рекомендации:

Изучить материалы лекций №№4-6.

Краткая теория

К операторам, позволяющим из нескольких возможных вариантов выполнения программы (ветвей) выбрать только один, относятся if и case.

Условный оператор if

Оператор if выбирает между двумя вариантами развития событий:

```
if <условие>
  then <один_оператор>
  [else <один_оператор>];
```

Обратите внимание, что перед словом **else** (когда оно присутствует) символ ";" не ставится – так как это разделит оператор на две части.

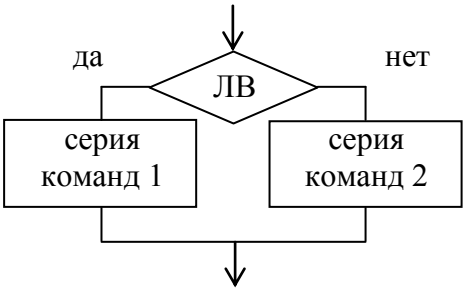
Условный оператор if работает следующим образом:

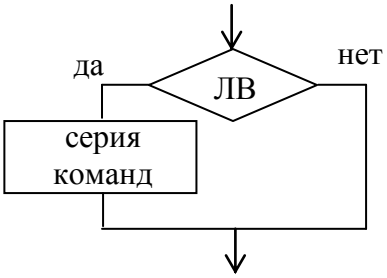
1. Сначала вычисляется значение <условия> - это может быть любое выражение, возвращающее значение типа boolean.

2. Затем, если в результате получена "истина" (true), то выполняется оператор, стоящий после ключевого слова then, а если "ложь" (false) - без дополнительных проверок выполняется оператор, стоящий после ключевого слова else. Если же else-ветвь отсутствует, то не выполняется ничего.

В случае, когда каждый оператор if имеет собственную else-ветвь, ошибки не будет. Но, если некоторые из операторов этой ветви не имеют, может возникнуть ошибка. Компилятор языка Pascal всегда считает, что else относится к самому ближнему оператору if. Для того чтобы избежать ошибок, необходимо всегда (или по крайней мере при наличии нескольких вложенных условных операторов) указывать оба ключевых слова, даже если одна из ветвей будет пустовать.

Изображение условного оператора в виде блок-схемы выглядит следующим образом: (слева – полное ветвление *если-то-иначе*, справа – неполный вариант ветвления *если-то*)





3. Порядок выполнения задания:

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первое задание; №по журналу +3 – второе задание и №по журналу +5 – третье задание (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Ознакомиться с условием задания и уяснить его.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

4. Варианты заданий:

1. Даны три целых числа. Возвести в квадрат отрицательные числа и в третью степень — положительные (число 0 не изменять).

2. Из трех данных чисел выбрать наименьшее.

3. Из трех данных чисел выбрать наибольшее.

4. Из трех данных чисел выбрать наименьшее и наибольшее.

5. Перераспределить значения переменных X и Y так, чтобы в X оказалось меньшее из этих значений, а в Y — большее.

6. Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.

7. Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения.

8. Даны три переменные: X, Y, Z. Если их значения упорядочены по убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

9. Даны три переменные: X, Y, Z. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное.

10. Даны целочисленные координаты точки на плоскости. Если точка не лежит на координатных осях, то вывести 0. Если точка совпадает с началом координат, то вывести 1. Если точка не совпадает с началом координат, но лежит на оси OX или OY, то вывести соответственно 2 или 3.

11. Даны вещественные координаты точки, не лежащей на координатных осях OX и OY. Вывести номер координатной четверти, в которой находится данная точка.

12. На числовой оси расположены три точки: A, B, C. Определить, какая из двух последних точек (B или C) расположена ближе к A, и вывести эту точку и ее расстояние от точки A.

13. Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Вывести порядковый номер этого числа.

14. Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.

15. Дан номер некоторого года (положительное целое число). Вывести число дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за ис-

ключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

16. Для данного x вычислить значение следующей функции f, вещественные значения: –1 если x <= , 0, f(x x· если 0 < x)= x, < 2, 4, если x >= 2.

17. Для данного x вычислить значение следующей функции f, принимающей значения целого типа: 0, если x < 0, f(x 1, если x принадлежит [0,1),)= [2,3), ... , –1 если x принадлежит [1,2), , [3,4), ...

18. Дано целое число, лежащее в диапазоне от –999 до 999. Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.

19. Дано целое число, лежащее в диапазоне от 1 до 9999. Вывести строку — словесное описание данного числа вида "четное двузначное число", "нечетное четырехзначное число" и т.д.

20. Два прямоугольника заданы длинами сторон. Определите, можно ли первый прямоугольник целиком разместить во втором.

21. Решите квадратное уравнение $ax^2 + bx + c = 0$

22. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данная тройка натуральных чисел a, b, c не является тройкой Пифагора, т.е. $c^2 = a^2 + b^2$

23. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данные числа x, y являются координатами точки, лежащей в первой координатной четверти.

24. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данная тройка натуральных чисел a, b, c является тройкой Пифагора, т.е. $c^2 = a^2 + b^2$.

25. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: число c является средним арифметическим чисел a и b.

26. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: число c является средним геометрическим чисел a и b.

27. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: сумма двух натуральных чисел кратна 2.

28. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: произведение натуральных чисел a и b кратно числу c.

29. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: данное натуральное число a кратно числу b, но не кратно числу c.

30. Составить программу, печатающую значение true, если указанное высказывание является истинным, и false в противном случае: числа a и b выражают длины катетов одного прямоугольного треугольника, c и d — другого. Эти треугольники являются подобными.

Методические указания к практическому занятию 3

Управляющие операторы СИ

1. Цели занятия: Практически освоить основные управляющие операторы языка. Практически потренироваться в их использовании при составлении программ.

2. Рекомендации:

Изучить материалы лекций №№7,8.

Краткая теория.

Управляющие операторы. Операторы перехода и циклов.

Оператор условного перехода If.

Оператор If... дает возможность в зависимости от условия выполнять ту или иную ветвь программы. Синтаксис оператора следующий:

If условие выражение1 else выражение2;

Условие должно давать результат в виде логического значения истинности или ложности. Выражение1 будет выполняться если условие истинно. Выражение2 будет выполняться если условие ложно.

Существует сокращенный вариант оператора:

If условие выражение1

Пример. Определить, является ли введенное число днем недели, т.е. входит ли число в диапазон от 1 до 7.

```
#include <stdio.h>
int             A;
main()
{
        printf("?  ");
        scanf("%d",&A);
        if ((A < 1) || (A > 7))
                printf("Error %d\n",A);
        else printf("OK %d\n",A);
```

```
}
```

Выражение условия (A<1) || (A>7) будет давать TRUE, если выполняется A<1 или A>7 - в этом случае выполняется ветка printf('Error ',A);, иначе ветка printf('OK ',A);.

Существует другой вариант записи оператора If ... Пример:

```
#include <stdio.h>

main()

{

        int y,t;

        printf("?  ");

        scanf("%d",&t);

        y=(t>0)? t*10: t-10; /* if t>0 y=t*10 else y=t-10;*/

        printf("OK %d\n",y);

}
```

В данном варианте вид оператора показан в комментариях.

Оператор switch... case множественного выбора из меню.

Оператор switch... case используется в случае, когда необходимо анализировать переменную и в зависимости от ее значения производить те или иные действия. Рассмотрим пример. С клавиатуры вводятся буквы латинского алфавиты. В зависимости от буквы произвести те или иные действия.

```
#include <stdio.h>

char        A;

main()

{

        printf("?  ");

        scanf("%c",&A);

        switch (A) {

                case 'c':   printf(" smoll %c\n",A); break; /* выход из блока */
```

```
            case 'F':
            case 'G':   printf(" big %c\n",A);
            break;
            default: printf("Error %c\n",A);
        }
}
```

В данном примере если введен символ c, то выполняется printf(" smoll %c\n",A);, если вводится заглавные буквы F или G, то выполняется printf(" big %c\n",A);, если не один из рассмотренных символов не вводится, то выполняется printf("Error %c\n",A);.

Оператор цикла с постусловием do... while

Для повторения некоторого множества команд несколько раз можно использовать оператор do... while. Рассмотрим пример.

```
#include <stdio.h>
main()
{
        int A;
        do {
            printf("Zifra?  ");
            scanf("%d",&A);
            printf("Error %d\n",A);
        }  while (!(A == 9));
        printf("OK %d\n",A);
}
```

С клавиатуры вводится число. Выполняется оператор printf("Error %d\n",A);. Далее идет анализ - равно число 9 или нет, если не равно, снова выполняется тело цикла:

printf("Zifra? ");

```
        scanf("%d",&A);
        printf("Error %d\n",A).
```

Если число равно 9, то выполняется оператор printf("OK %d\n",A); и работа цикла заканчивается.

Главной особенностью оператора do... while является тот факт, что тело цикла, заключенное между операторами do и while выполняется хотя бы один раз, т.е. вначале выполняется тело цикла, а затем идет анализ условия.

Таким образом, смысл рассматриваемого оператора заключается в следующем: "Выполняй тело цикла до тех пор, пока истинно условие".

Оператор цикла с предусловием while

Оператор while... в отличие от do... while вначале анализирует условие, а затем выполняет тело цикла.

Пример.

```
#include <stdio.h>
main()
{
    int A;
    A = 0;
    while (A != 9)
    {
        printf("Zifra? ");
        scanf("%d",&A);
        printf("Error %d\n",A);
    }
    printf("OK %d\n",A);
}
```

В данном примере инициализирована переменная A:=0;. Это сделано, потому что вначале идет анализ равна она 9 или нет. Если не равна, то выполняется тело цикла. Смысл рассматриваемого оператора заключается в следующем:

«Пока истинно условие выполняй тело цикла».

Оператор цикла с параметром for

Оператор for... используется когда известно сколько раз необходимо выполнить тело цикла, но этот оператор горазда гибче по сравнению с Паскалем. Рассмотрим пример.

```
#include <stdio.h>

int                 A;

main()

{

        for (A = 1; A <= 5; A++) /* A++ означает A=A+1 */

        printf("Zifra %d\n",A);

}
```

В этом примере A хранит состояние счетчика цикла. Первоначально A:=1. Выполняется оператор printf("Zifra %d\n",A). Далее значение A увеличивается на единицу. Идет анализ A<=5 или нет. Если A больше 5, то цикл заканчивает работу. Если нет, то снова выполняется оператор printf("Zifra %d\n",A).

В следующем примере рассмотрим вариант оператора for..., когда начальное значение переменной больше конечного, а переменная во время работы цикла уменьшается на единицу.

```
#include <stdio.h>

int         A;

main()

{

        for (A = 5; A >= 1; A--) /* A-- означает A=A-1 */

        printf("Zifra %d\n",A);

}
```

Существует множество модификаций оператора for..., например:

- пустой оператор - для временной задержки:

```
for (n=1;n <=10000;n++)

; /* пустой оператор */
```

- использование различного шага:

```
for (n=1;n <=230;n=n+10)
```

- изменение переменных:

```
for (x=2;n*n <=476;n=5*x++)
```

Рассмотрим пример, в котором инициализируются две переменные и каждая из которых, изменяется после итерации цикла:

```
#include <stdio.h>

#define f 30

#define n 19

main()

{

    int y,t;

    for (y = 1,t=f; y<=16; y++,t+=n) /*t+=n означает t=t+n*/ printf(" %3d %7d\n",y,t);

}
```

3. Порядок выполнения задания:

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первое задание; №по журналу +3 – второе задание и №по журналу +5 – третье задание (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Ознакомиться с условием задания и уяснить его.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

4. Варианты заданий:

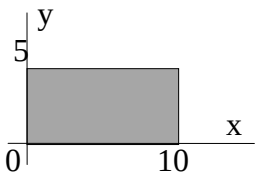
Решить задачи, используя оператор условного перехода if

1. Даны действительные числа A,B,C,D. Выяснить, можно ли уместить прямоугольник со сторонами A,B внутри прямоугольника со сторонами C,D.

2. Даны действительные числа x, y, z. Найти минимальное из них.

3. Даны действительные положительные числа A,B,C. Выяснить, пройдет ли кирпич с ребрами A,B,C в прямоугольное отверстие со сторонами x,y.

4.Определить, лежит ли точка D(c,b), где $c=\sqrt{a1+a2},\ b=a1+0{,}7a3,$ внутри прямоугольника (a1,a2,a3 - произвольные числа)



5. Выяснить, существует ли треугольник с координатами вершин A(x1,y1), B(x2,y2), C(x3,y3), если да, то найти его площадь.

6. Даны действительные числа A,B,C. Проверить выполняются ли неравенства A<B<C, если да, то присвоить A=B+C иначе A=C-B.

7. Даны действительные числа x,y. Вычислить значение функции z=log(x-y)-x/y

8. На плоскости расположена окружность радиуса R с центром в начале координат
Определить положение точки x с координатами (A,B) относительно окружности.

9. Даны круг радиуса R и квадрат со стороной A. Определить их взаимное положение.

10. Вывести на печать переменные A,B,C в порядке их возрастания

11. Проверить, какие из чисел A,B,C,D принадлежат интервалу (1,25).

12. Даны действительные числа A,B. Если они оба отрицательные, то заменить каждое из них его квадратом, иначе положительные из них увеличить в два раза.

13. Выяснить, существует ли треугольник с координатами вершин A(x1,y1), B(x2,y2),C(x3,y3).

14. Даны действительные числа x,y. Вычислить значение функции z=log(x/y)-1/x

15. Даны действительные числа A,B. Если они оба неотрицательные, то заменить каждое из них его кубом, иначе отрицательные из них заменить их модулями.

16. Даны площадь квадрата S1 и круга S2. Определить поместится ли круг в квадрат и наоборот.

17. На плоскости расположена окружность радиуса R с центром в начале координат. Определить, лежат ли точки A(x1,y1) и B(x2,y2) на окружности.

18. Составить программу вычисления корней системы уравнений с двумя неизвестными $\begin{cases} a1x + b1y = c1 \\ a2x + b2y = c2 \end{cases}$ методом Крамера. Прверить, что главный определитель не равен 0.

19. Даны действительные числа A,B,C,D. Выяснить, можно ли уместить прямоугольник со сторонами A,B внутри прямоугольника со сторонами C,D.

20. Вывести на печать переменные A,B,C в порядке их убывания

21. Даны действительные числа x, y, z. Найти максимальное из них.

22. Проверить, какие из чисел A,B,C,D не принадлежат интервалу (3,15).

23. Даны действительные числа x,y. Вычислить значение функции z=ln(x)-x/y

24. Даны действительные числа A,B. Если они имеют разные знаки, то напечатать их произведение, иначе напечатать их квадраты

25. Выяснить, существует ли треугольник с длинами сторон A,B,C. Если да, то найти его площадь.

26. Даны действительные числа x,y. Вычислить значение функции z=arcsin(x)-y

27. Даны действительные числа x,y,z. Получить максимальное из них по модулю

28. Даны действительные числа x,y. Вычислить значение функции z=arcsin(x+y)

29. На каком из интервалов $(-\infty;k1),(k1;k2),(k2;+\infty)$ лежит точка с координатой x?
k1, k2, x - произвольные числа, причем k1<k2.

30. Лежат ли обе точки D(a1;b1) и C(a2;b2) внутри круга радиуса R с центром в начале координат? Если такой точки нет, выдать соответствующее сообщение.

Решить задачи, используя все виды циклов

1. Найти среднее арифметическое положительных чисел, введенных с клавиатуры. Всего ввести N различных чисел.

2. Ввести с клавиатуры N чисел. Найти сумму тех из них, которые принадлежат интервалу (2;9).

3. Для N введенных с клавиатуры чисел найти сумму положительных кратных 3.

4. Для арифметической прогрессии 4, 9, 14, 19... найти первые n членов этой прогрессии.

5. Найти сумму отрицательных значений функции Z=sin(5-x)/cos(x-2) для x, изменяющегося на отрезке [-5,12] с шагом 1.

6. Найти среднее арифметическое отрицательных чисел, введенных с клавиатуры. Всего ввести N различных чисел.

7. Найти среднее арифметическое чисел, принадлежащих отрезку [2,184], кратных 2 и введенных с клавиатуры. Всего ввести N различных чисел.

8.Найти сумму значений функции, больших 2 Z=sin(1/x)+5cos(1/(x-3))+x для x, изменяющегося на отрезке [-3,8] с шагом 1.

9.Найти n членов последовательности $x_1 = x_2 = x_3 = 1$;
$x_k = x_{k-1} + x_{k-3}$.

10. Вычислить последовательность N чисел $A_0 = x$,
$A_1 = 2$, $A_k = A_{k-1} - A_{k-2}$.

11. Для $x_1 = 0,3$ и $x_2 = -0,3$ найти $x_k = k + \sin(x_{k-2})$ для k, изменяющегося следующим образом: k = 3,4,...,14.

12. Составить таблицу перевода дюймов в сантиметры для расстояний от 1 до 13 дюймов с шагом 1.

13. Вывести на печать значения функции, меньшие 2, Z=sin(1/x)+5cos(x-3)+x для x, изменяющегося на отрезке [-7,4] с шагом 1.

14.Напечатать таблицу значений функции
Y = tg(x/b)+x/(b-2) для x, изменяющегося от 0 до 10 с шагом 1 (b - произвольное число).

15. Вычислить N -ый член последовательности
$x_k = x_{k-2} - x_{k-1}$, $x_0 = 2,4$ $x_1 = 3,8$.

16. Составить таблицу перевода суток (от 1 до 7) в часы, минуты, секунды.

17. Вычислить N-ый член последовательности
$x_k = x_{k-1} + (2/3)x_{k-2} + 1$, $x_1 = -1$, $x_2 = 1,38$.

18. Напечатать значения функции
z = 1/(x-2)+1/(x-5)+ln(12,8-X) для x, изменяющегося на отрезке [-4,14] с шагом 1.

19. Вывести на печать отрицательные значения функции z=sin(5-x)/cos(x-2) для x, изменяющегося на отрезке [-6,13] с шагом 1 (учесть область допустимых значений функции).

20. Из N введенных с клавиатуры чисел напечатать кратные 3 и меньшие 58.

21. Ввести с клавиатуры N чисел. Напечатать те из них, которые принадлежат интервалу (1,11) и являются четными.

22. Из N введенных с клавиатуры чисел напечатать положительные, кратные 3.

23. Вывести на печать значения функции
z = sin(x/(x-2)), находящиеся в интервале (-0,4;0,8) для x, изменяющегося от 8 до -6 с шагом 1.

24. Ввести с клавиатуры N чисел. Напечатать те из них, которые принадлежат интервалу (2;9).

25. Для геометрической прогрессии 2, 6, 18, 54, 162 ... определить первые n членов этой прогрессии.

26. Ввести с клавиатуры N чисел. Напечатать те из них, которые не принадлежат интервалу (1;5).

27. Найти n членов последовательности $x_1 = x_2 = x_3 = 1$; $x_k = x_{k-1} - 2x_{k-3}$.

28. Вычислить последовательность N чисел $A_0 = x$,
$A_1 = 2$, $A_k = A_{k-1} + A_{k-2}$

29. Составить таблицу перевода килограммов (от 1 до 13) в граммы с шагом

30. Найти сумму значений функции Y=COS(x/A)+x/(A-2) для x, изменяющегося от 2 до 13 с шагом 1 (A - произвольное число).

Методические указания к практическому занятию 4

Решение задач в Java

1. Цель занятия:

Выработать умения и навыки составлять типовые программы решения задач на выбранном языке программирования.

2. Рекомендации:

Изучить материалы лекций №№12-14.

Краткая теория

Базовым элементом объектно-ориентированного программирования в языке Java является класс. В этой главе Вы научитесь создавать и расширять свои собственные классы, работать с экземплярами этих классов и начнете использовать мощь объектно-ориентированного подхода. Напомним, что классы в Java не обязательно должны содержать метод main. Единственное назначение этого метода — указать интерпретатору Java, откуда надо начинать выполнение программы. Для того, чтобы создать класс, достаточно иметь исходный файл, в котором будет присутствовать ключевое слово class, и вслед за ним — допустимый идентификатор и пара фигурных скобок для его тела.

```
class Point {
}
```

ЗАМЕЧАНИЕ
Имя исходного файла Java должно соответствовать имени хранящегося в нем класса. Регистр букв важен и в имени класса, и в имени файла.

Класс — это шаблон для создания объекта. Класс определяет структуру объекта и его *методы,* образующие функциональный интерфейс. В процессе выполнения Java-программы система использует определения классов для создания представителей классов. Представители являются реальными *объектами.* Термины «представитель», «экземпляр» и «объект» взаимозаменяемы. Ниже приведена общая форма определения класса.

```
class имя_класса extends имя_суперкласса { type переменная1_объекта:
type переменная2_объекта:
type переменнаяN_объекта:
type имяметода1(список_параметров) { тело метода;
}
type имяметода2(список_параметров) { тело метода;
}
type имя методаM(список_параметров) { тело метода;
}
}
```

Ключевое слово extends указывает на то, что *«имя_класса»* — это подкласс класса *«имя_суперкласса».* Во главе классовой иерархии Java стоит единственный ее встроенный класс — Object. Если вы хотите создать подкласс непосредственно этого класса, ключевое слово extends и следующее за ним имя суперкласса можно опустить — транслятор включит

их в ваше определение автоматически. Примером может служить класс Point, приведенный ранее.

Переменные представителей (instance variables)
Данные инкапсулируются в класс путем объявления переменных между открывающей и закрывающей фигурными скобками, выделяющими в определении класса его тело. Эти переменные объявляются точно так же, как объявлялись локальные переменные в предыдущих примерах. Единственное отличие состоит в том, что их надо объявлять вне методов, в том числе вне метода main. Ниже приведен фрагмент кода, в котором объявлен класс Point с двумя переменными типа int.

```
class Point { int x, y;
}
```

В качестве типа для переменных объектов можно использовать как любой из простых типов так и классовые типы. Скоро мы добавим к приведенному выше классу метод main, чтобы его можно было запустить из командной строки и создать несколько объектов.

Оператор new
Оператор new создает экземпляр указанного класса и возвращает ссылку на вновь созданный объект. Ниже приведен пример создания и присваивание переменной p экземпляра класса Point.

```
Point p = new Point();
```

Вы можете создать несколько ссылок на один и тот же объект. Приведенная ниже программа создает два различных объекта класса Point и в каждый из них заносит свои собственные значения. Оператор точка используется для доступа к переменным и методам объекта.

```
class TwoPoints {
public static void main(String args[]) {
Point p1 = new Point();
Point p2 = new Point();
p1.x = 10;
p1.y = 20;
p2.x = 42;
p2.y = 99;
System.out.println("x = " + p1.x + " y = " + p1.y);
System.out.println("x =  " + p2.x + " y = " + p2.y);
} }
```

В этом примере снова использовался класс Point, было создано два объекта этого класса, и их переменным x и y присвоены различные значения. Таким образом мы продемонстрировали, что переменные различных объектов независимы на самом деле. Ниже приведен результат, полученный при выполнении этой программы.

```
C:\> Java TwoPoints
x = 10 y = 20
x = 42 y = 99
```

ЗАМЕЧАНИЕ
Поскольку при запуске интерпретатора мы указали в командной строке не класс Point, а класс TwoPoints, метод main класса Point был полностью проигнорирован. Добавим в класс Point метод main и, тем самым, получим законченную программу.

```
class Point { int x, y;
public static void main(String args[]) {
```

```
Point p = new Point();
p.x = 10;
p.y = 20;
System.out.println("x =  " + p.x + " y = " + p.y);
} }
```

Объявление методов

Методы - это подпрограммы, присоединенные к конкретным определениям классов. Они описываются внутри определения класса на том же уровне, что и переменные объектов. При объявлении метода задаются тип возвращаемого им результата и список параметров. Общая форма объявления метода такова:

```
тип имя_метода (список формальных параметров) {
тело метода:
}
```

Тип результата, который должен возвращать метод может быть любым, в том числе и типом void - в тех случаях, когда возвращать результат не требуется. Список формальных параметров - это последовательность пар тип-идентификатор, разделенных запятыми. Если у метода параметры отсутствуют, то после имени метода должны стоять пустые круглые скобки.

```
class Point { int x, y;
void init(int a, int b) {
x = a;
У = b;
} }
```

Вызов метода

В Java отсутствует возможность передачи параметров *по ссылке* на примитивный тип. В Java все параметры примитивных типов передаются *по значению,* а это означает, что у метода нет доступа к исходной переменной, использованной в качестве параметра. Заметим, что все объекты передаются по ссылке, можно изменять содержимое того объекта, на который ссылается данная переменная.

Скрытие переменных представителей

В языке Java не допускается использование в одной или во вложенных областях видимости двух локальных переменных с одинаковыми именами. Интересно отметить, что при этом не запрещается объявлять формальные параметры методов, чьи имена совпадают с именами переменных представителей. Давайте рассмотрим в качестве примера иную версию метода init, в которой формальным параметрам даны имена x и y, а для доступа к одноименным переменным текущего объекта используется ссылка **this**.

```
class Point { int x, y;
void init(int x, int y) {
this.x = x;
this.y = y } }
class TwoPointsInit {
public static void main(String args[]) {
Point p1 = new Point();
Point p2 = new Point();
p1.init(10,20);
p2.init(42,99);
System.out.println("x = " + p1.x + " y = •• + p-l.y);
System.out.printlnC'x = " + p2.x + " y = •• + p2.y);
} }
```

Конструкторы

Инициализировать все переменные класса всякий раз, когда создается его очередной представитель — довольно утомительное дело даже в том случае, когда в классе имеются функции, подобные методу init. Для этого в Java предусмотрены специальные методы, называемые конструкторами. Конструктор — это метод класса, который инициализирует новый объект после его создания. Имя конструктора всегда *совпадает* с именем класса, в котором он расположен (также, как и в C++). У конструкторов нет типа возвращаемого результата - никакого, даже void. Заменим метод init из предыдущего примера конструктором.

```
class Point { int x, y;
Point(int x, int y) {
this.x = x;
this.y = y;
} }
class PointCreate {
public static void main(String args[]) {
Point p = new Point(10,20);
System.out.println("x = " + p.x + " y = " + p.y);
} }
```

Программисты на Pascal (Delphi) для обозначения конструктора используют ключевое слово **constructor**.

Совмещение методов

Язык Java позволяет создавать несколько методов с одинаковыми именами, но с разными списками параметров. Такая техника называется совмещением методов (**method overloading**). В качестве примера приведена версия класса Point, в которой совмещение методов использовано для определения альтернативного конструктора, который инициализирует координаты x и y значениями по умолчанию (-1).

```
class Point { int x, y;
Point(int x, int y) {
this.x = x;
this.y = y;
}
Point() {
x = -1;
y = -1;
} }
class PointCreateAlt {
public static void main(String args[]) {
Point p = new Point();
System.out.println("x = " + p.x + " y = " + p.y);
} }
```

В этом примере объект класса Point создается не при вызове первого конструктора, как это было раньше, а с помощью второго конструктора без параметров. Вот результат работы этой программы:

C:\> java PointCreateAlt

x = -1 y = -1

ЗАМЕЧАНИЕ

Решение о том, какой конструктор нужно вызвать в том или ином случае, принимается в соответствии с количеством и типом параметров, указанных в операторе new. Недопустимо объявлять в классе методы с одинаковыми именами и сигнатурами. В сигнатуре метода не учитываются имена формальных параметров учитываются лишь их типы и количество.

3. Порядок выполнения задания

3.1. Выбрать 3 варианта задания из перечня вариантов, приведенных ниже по следующему правилу: №по журналу- первое задание; №по журналу +3 – второе задание и №по журналу +5 – третье задание (если достигнуто окончание списка вариантов зданий, то перейти в его начало).

3.2. Ознакомиться с условием задания и уяснить его.

3.3. Составить программу по заданию.

3.4. Оттранслировать программу на изучаемом языке программирования и получить решение задачи.

3.5. Оформить отчет для каждой из 3 задач, включив в него задание, блок-схему алгоритма (в электронном виде), текст программы и skrin-shert результата выполнения задания и представить его на проверку.

4. Варианты заданий:

Составить функции пользователя (методы) для следующих задач.

1. Составить программу для перевода длины в метрах в длину в сантиметрах, определив функцию, выполняющую это преобразование и передав длину в метрах в качестве параметра.

2. Составить программу для нахождения суммы элементов каждого из трех массивов, введенных с клавиатуры, определив функцию, выполняющую это действие, и передавая массивы в качестве параметра.

3. Даны числа S, T. Получить с использованием функции пользователя F(T,-2S,1.17)+F(2.2,T,S-T) где F(A, B, C) = (2A-B-sin(C))/(5+C)

4. Составить программу перевода двоичной записи натурального числа в десятичную, описав соответствующую функцию с параметром. Перевод осуществлять для чисел, вводимых с клавиатуры. Признак конца ввода - число 0.

5. Даны числа S, T. Получить с использованием функции пользователя с параметрами G(1,sin(S))+2G(T*S,24)-G(5,-S), где G(A,B)=(2A+B*B)/(A*B*2+B*5).

6. Составить программу для расчета значений гипотенузы треугольника, определив функцию, выполняющую этот расчет. Катеты передаются в качестве параметров.

7. Найти периметр десятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, которые передаются функции в качестве параметров из основной программы.

8. Найти периметр шестиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами. Координаты передаются функции в качестве параметров из основной программы.

9. Найти площадь пятиугольника, координаты вершин которого заданы. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами, и процедуру вычисления площади треугольника по трем сторонам. Описать функции с соответствующими формальными параметрами.

10. Составить программу вывода на экран всех натуральных чисел, не превосходящих N и делящихся на каждую из своих цифр. Описать соответствующую функцию, получающую из основной программы в качестве параметра натуральное число и возвращающую TRUE, если оно удовлетворяет указанному условию.

11. Используя подпрограмму - функцию, составить программу для нахождения максимального из трех чисел. Числа передаются функции в качестве параметров.

12. Используя подпрограмму - функцию, составить программу для печати знаков трех чисел, введенных с клавиатуры и передаваемых функции в качестве параметра.

13. Используя подпрограмму - функцию, составить программу для возведения чисел в целую положительную степень. Число передаются функции в качестве параметра из основной программы. Расчет вести для чисел, пока не будет введено число, равное 0.

14. Используя подпрограмму - функцию, составить программу для вычисления функции Z=(X1+Y1)/(X1*Y1), где X1 - первый корень уравнения X^2-4*X-1=0; Y1 - первый корень уравнения $2*Y^2 + A*Y - A^2 = 0$ (A - произвольное).

15. Задав функцию, вывести на печать средние арифметические двух массивов, введенных с клавиатуры. Массив передается функции в качестве параметра.

16. Задав функцию, рассчитать и вывести на печать максимальные значения в трех парах чисел, вводимых с клавиатуры. Пара чисел передается функции в качестве параметра.

17. Найти периметр восьмиугольника, координаты вершин которого заданы. Определить функцию вычисления расстояния между двумя точками, заданными своими координатами. Координаты передать функции в качестве параметров.

18. Даны четыре пары чисел. Получить с использованием функции пользователя наибольший общий делитель для каждой пары.

19. Даны числа A, B, C. Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров.

20. Даны числа x = 1,2,...,N. Получить с использованием функции пользователя значения 3*P(X+3)*P(X) для заданных x, где $P(X) = 10*X^3 - 14*X^2 + 12*X - 2$.

21. Составить программу для расчета значений катета треугольника, определив функцию, выполняющую этот расчет. Гипотенуза и второй катет передаются в качестве параметров.

22. Даны целые числа a,b,c,d. Проверить с использованием функции пользователя их четность. Число для проверки передается в функцию в качестве параметра из основной программы.

23. Для каждого из 10 введенных с клавиатуры чисел напечатать сообщение: является ли оно простым или нет, описав функцию логического типа, возвращающую значение “ИСТИНА”, если число, переданное ей в качестве параметра, является простым.

24. Даны числа S, T. Получить с использованием функции пользователя Y(T,S)=G(12,S)+G(T,S)-G(2S-1,S*T), где G(A,B)=(2*A+B*B)/(A*B*2+B*5).

25. Определите функцию, определяющую, какой целой степенью числа 2 является ее аргумент (если число не является степенью двойки - выдать соответствующее сообщение).

26. Определите функцию, подсчитывающую сумму N первых элементов целочисленного массива A. N и массив A передать в качестве параметров.

27. Вычислить количество простых чисел, не превосходящих заданного N. Описать функцию логического типа, возвращающую значение true, если число простое и false в противном случае.

28. Используя подпрограмму - функцию с параметрами, составить программу для вычисления функции $F(X,Y) = (2X^3-4*X^2+X+1)/(9*Y^3+Y+4) + 3*Y^2+5*Y$.

29. Составить программу для перевода веса в граммах в вес в килограммах, определив функцию, выполняющую это преобразование. Вес в граммах передается функции в качестве параметра.

30. Даны числа S, T. Получить с использованием функции пользователя G(12, S)+G(T, S)-G(2S-1, S*T) где G(A, B) = (2*A+B*B)/(A*B*2+B*5).