ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

Утверждаю
Зам. директора по УР
_____ Жуковский А. Г.
« 28 » 08 2019 г.

Б1.В.ДВ.10.02 ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

рабочая программа дисциплины

Кафедра **«Информатика и вычислительная техника»**
Направление подготовки **09.03.01. Информатика и вычислительная техника**
Профиль **Программное обеспечение и интеллектуальные системы**

Формы обучения **очная, заочная**

Распределение часов дисциплины по семестрам (ОФ), курсам (ЗФ)

Вид учебной работы	ОФ		ЗФ	
	ЗЕ	часов	ЗЕ	часов
Общая трудоемкость дисциплины, в том числе (по семестрам, курсам):	3	108/7	3	108/4
Контактная работа, в том числе (по семестрам, курсам):		64/7		14/4
Лекции		32/7		6/4
Лабораторных работ		16/7		4/4
Практических занятий		16/7		4/4
Семинаров				
Самостоятельная работа		17/7		94/4
Контроль		27		
Число контрольных работ (по курсам)				
Число КР (по семестрам, курсам)				
Число КП (по семестрам, курсам)				
Число зачетов с разбивкой по семестрам				
Число экзаменов с разбивкой по семестрам (курсам)		1/7		1/4

Программу составил:
Старший преподаватель кафедры ИВТ Конева С. И.

Рецензент(ы):
Профессор кафедры ИВТ, д. т. н., профессор Соколов С. В.

Рабочая программа дисциплины
«Функциональное программирование»

Разработана в соответствии с ФГОС ВО:
ФЕДЕРАЛЬНЫЙ ГОСУДАРСТВЕННЫЙ ОБРАЗОВАТЕЛЬНЫЙ СТАНДАРТ ВЫСШЕГО ОБРАЗОВАНИЯ
Направление подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
УТВЕРЖДЕН Приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 929

Составлена на основании учебных планов
направления 09.03.01 Информатика и вычислительная техника профиля «Программное обеспечение и интеллектуальные системы», одобренных Учёным советом СКФ МТУСИ, Протокол № 5 от 24.12.2018, и утвержденных директором СКФ МТУСИ 15.01.2019 г.

Одобрена на заседании кафедры
"Информатика и вычислительная техника"

Протокол от 26.8.19 № 1
Зав. кафедрой /Соколов С. В./

Визирование для использования в 20__/20__ уч. году

Утверждаю
Зам. директора по УВР_____________________
« __» _______ 20__ г.
Рабочая программа пересмотрена, обсуждена и одобрена на заседании кафедры
« Информатика и вычислительная техника»

Протокол от « __» ________ 20__ г. № _
Зав. кафедрой _____________________

Визирование для использования в 20__/20__ уч. году

Утверждаю
Зам. директора по УВР_____________________
« __» _______ 20__ г.
Рабочая программа пересмотрена, обсуждена и одобрена на заседании кафедры
« Информатика и вычислительная техника»

Протокол от «__» ________ 20__ г. № _
Зав. кафедрой _____________________

Визирование для использования в 20__/20__ уч. году

Утверждаю
Зам. директора по УВР_____________________
« __» _______ 20__ г.
Рабочая программа пересмотрена, обсуждена и одобрена на заседании кафедры
« Информатика и вычислительная техника»

Протокол от «__» ________ 20__ г. № _
Зав. кафедрой _____________________

Визирование для использования в 20__/20__ уч. году

Утверждаю
Зам. директора по УВР_____________________
« __» _______ 20__ г.
Рабочая программа пересмотрена, обсуждена и одобрена на заседании кафедры
« Информатика и вычислительная техника»

Протокол от «__» ________ 20__ г. № _
Зав. кафедрой _____________________

---

1. Цели изучения дисциплины

Целями освоения дисциплины «Функциональное программирование » является изучение теоретических и практических основы функционального программирования, языка Haskell, используемого как для представления «типичного» языка функционального программирования, так и для практического применения в сложных многопроцессорных и распределённых системах.

2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у выпускника способность решать следующие профессиональные задачи в соответствии с проектным видом деятельности.

Результатом освоения дисциплины являются сформированные у выпускника следующие компетенции:

Компетенции выпускника, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)
ПК-1. Способность разрабатывать требования и проектировать программное обеспечение
Знать:
Методы и приёмы формализации, алгоритмизации, программирования и оформления программного кода; Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними; Методологии и технологии проектирования и использования баз данных; Основные методы измерения и оценки характеристик программного обеспечения.
Уметь:
Разрабатывать программное обеспечение с использованием языков и сред программирования, выполнять определение и манипулирование данными; Осуществлять тестирование, отладку и оптимизацию программного обеспечения; Использовать выбранную среду программирования для разработки процедур интеграции программных модулей.
Владеть:
Приёмами анализа возможностей и разработки требований к программному обеспечению; Методами проектирования программного обеспечения и баз данных; Методами и средствами интеграции модулей и компонент программного обеспечения, приёмами развёртывания и обновления программного обеспечения.

3. Место дисциплины в структуре образовательной программы

Требования к предварительной подготовке обучающегося (предшествующие дисциплины, модули, темы):	
1	Б1.В.02 Методы и средства проектирования информационных систем
2	Б1.В.06 Дискретная математика
3	Б1.В.ДВ.07.02 Интеллектуальные информационные системы
Последующие дисциплины и практики, для которых освоение данной дисциплины необходимо:	
1	Б1.В.ДВ.12.01 Проектирование клиент – серверных приложений
2	Б1.В.ДВ.12.02 WEB - программирование

4. Структура и содержание дисциплины

4.1 Очная форма обучения, (всего 108 часов , 64 аудиторных часа)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компе-тенции	УМИО
1	2	3	4	5	6
Курс 4, Семестр 7					
Модуль 1:Концепция функционального программирования. - 36(32 +4СР)					
1.1	История функционального программирования: – общее представление о функциональном программировании (**ФП**) и его применении; – классификация языков функционального программирования; – история создания и развития языка **Haskell.**	Лек.	2	ПК-1	Л1.1
1.2	Недостатки императивного стиля программирования (принцип последовательного исполнения, влияние на скорость работы центрального процессора).	СР	1	ПК-1	Л1.1
1.3	Два способа проектирования вычислительной системы с параллельным вычислением, достоинства и недостатки.	СР	1	ПК-1	Л1.1
1.4	Математические основы **ФП - λ(лямда)** исчисление Алозно Черча: – основные понятия λ- исчисления; – место и роль различных редукций в λ-исчислении; – процесс преобразования формул в λ-исчислении; – нормальный порядок редукций; – методы представления констант и функций в чистом λ-исчислении.	Лек.	4	ПК-1	Л1.1
1.5	**Рекурсия** в λ-исчислении. Приведение выражения к нормальной и слабой заголовочной нормальной форме в системе редукций λ-исчисления.	СР	1	ПК-1	Л1.1
1.6	Упражнения по λ-исчислению. λ-исчисление как универсальный язык программирования	ПЗ1	2	ПК-1	Л1.1
1.7	Функциональный стиль программирования: – особенности функционального стиля; – отличительные особенности функций языков программирования в ФП; – «чистые» и «нечистые» функции, отсутствие присваиваний; – использование ***рекурсии*** вместо циклов	Лек	2	ПК-1	Л1.1

1.8	Представление функций в виде суперпозиций обращений к другим функциям в ФП.	СР	1	ПК-1	Л1.1
1.9	Решение задач на языке Java или Паскаль в императивном стиле и с использованием функционального стиля: – вычисление числа ***e*** в традиционном и функциональном стиле; – нахождение корня функции методом ***бисекции*** в традиционном и функциональном стиле; – вычисление суммы элементов массива в традиционном и функциональном стиле.	ПЗ2	4	ПК-1	Л1.1
1.10	Система программирования Haskell Platform: – установка системы программирования Haskell Platform; – запуск интерпретатора выражений на языке Haskell – GHCi (интерпретатор командной строки) или (в системе Windows) WinGHCi. GHC (Glasgow Haskell Compiler) – компилятор программ на языке; – получение и просмотр полного списка команд; – установление различных режимов работы интерпретатора; – проверка типа выражений, которые подлежат вычислению; – вычисление простых и сложных выражений; – загрузка определения функций для их последующего вызова (применения)	Лаб.1	2	ПК-1	Л1.1
1.11	Подготовка, загрузка и исполнение простых программы на языке Haskell с помощью установленного интерпретатора.	Лаб.2	4	ПК-1	Л1.1
1.12	Haskell – строго типизированный язык. Система типов языка Haskell: – целые с двумя подтипами(**Integer, Int**); – вещественные с двумя подтипами (**Float, Double**); – логические Bool (**True, False**); – символьные Char (выделение апострофами). Идентификаторы встроенных или определённых программистом типов, классов, модулей и пакетов. Идентификаторы объектов (простых и сложных типов, функций). Применение апострофов для построения имён	Лек.	4	ПК-1	Л1.1

	функций. Собственные идентификаторы.				
1.13	Элементарные типы данных. Полный список разрешенных операций для класса **Num.** Функция **fromInteger** для явного преобразования типов. Объединение типов в кортежи. Стандартные функции и операции. Две особенности записи выражений	ПЗ3	4	ПК-1	Л1.1
1.14	Определение функций с помощью уравнений: – задание типа функции и уравнения определения функции; – редукция–преобразование выражений. Осуществление вычисления выражений в Haskell (исполнение программы) с помощью последовательных редукций исходного выражения Приведение исходного выражения к ***нормальной форме.***	Лек.	4	ПК-1	Л1.1
	Модуль 2: Техника работы со списками, функции высших порядков. **Час** 45(32+13СР)				
2.1	Редукция выражений, работа которых не определяется уравнениями, содержащих примитивные встроенные операции (арифметические операции). ***Рекурсивно*е** определение функции. Вычисление условного выражения	ПЗ4	4	ПК-1	Л1.1
2.2	Определение функций с помощью уравнений; вычисление выражений в **Haskell** (исполнение программы) с помощью последовательных редукций исходного выражения; определение простой рекурсивной функции.	Лаб.3	4	ПК-1	Л1.1
2.3	Навыки написания программ на языке Haskell; технологии обработки списковых структур; техника преобразования рекурсивных функций в функции с концевой рекурсией**.**	Лаб.4	4	ПК-1	Л1.1
2.4	***Концевая рекурсия и*** *накапливающие аргументы*. Нахождение чисел Фибоначчи с помощью простого рекурсивного определения. Эффективное вычисление чисел Фибоначчи. Приближенное вычисление числа ***е*** - пример функции с накапливающими аргументами	Лек.	2	ПК-1	Л1.1
2.5	Вычисление факториала числа с помощью ***концевой рекурсии***	СР	1	ПК-1	Л1.1

2.6	Техника работы со списками: Определение и обозначение типа списка. Задание объектов-списков перечислением. Два полезных сокращения для записи списков. Конструктор списков. Операции обработки списков ***(head, last, tail, !!, null, length,*** ++). Суммирование элементов списка. Обращение списка.	Лек	4	ПК-1	Л1.1
2.7	Обработка списков строк, элементами которых служат списки символов. Функция проверки наличия в списке «белых» строк	СР	2	ПК-1	Л1.1
2.8	Написать функцию (программу), определяющую количество строк в списке, содержащих буквы.	Ср	2	ПК-1	Л1.1
2.9	Написать функцию сортировки списка методом «простых вставок» с помощью заголовка модуля (**module** Sort (simleSort) **where**).Вариант функции *simleSort* и функции insert с концевой рекурсией. Сравнение результатов с точки зрения замедления работы	СР	2	ПК-1	Л1.1
2.10	Подготовка функций (программ): – Метод слияния. Функция сортировки списка слиянием (сортировка фон Неймана) – Два варианта функций, вычисляющих число сочетаний из ***n*** предметов по ***m.*** Сравнение по быстродействию.	СР	2	ПК-1	Л1.1
2.11	Написать функцию (программу), вычисляющую длину самой длинной неубывающей последовательности подряд идущих чисел в заданном списке элементов	СР	2	ПК-1	Л1.1
2.12	Понятие *функций* ***высшего порядка***, область их применения в функциональных программах. (foldl, foldr) – ***операции свёртки списка***. Обработка списков с помощью функций высших порядков. Использование функций фильтрации (condition, filtered, quicksort). Использование функций высших порядков при обработке сложных структур	Лек	4	ПК-1	Л1.1
2.13	Использование функций высших порядков при обработке сложных структур данных. Функции отображения и свёртки.	ПЗ5	2	ПК-1	Л1.1
2.14	Преобразование карринговых функций в некарринговые и обратно.	Лек	4	ПК-1	Л1.1

	Функциональное представление данных. Представление множеств характеристическими функциями				
2.15	Навыки применения функций высших порядков в программировании на языке Haskell; навыки избавления от рекурсии при обработке сложных структур данных	Лаб.5	2	ПК-1	Л1.1
2.16	Класс отображаемых структур данных Functor. Моноид в языке Haskell (класс Monoid в пакете Data.Monoid). Значение и область применимости этих классов. Определение нейтрального элемента (***mempty***), бинарной операции (***mconcat***), соединение элементов списка моноидных значений с помощью операции (***mappend***).	СР	2	ПК-1	Л1.1
2.17	Ввод-вывод. Компиляция программ на Haskell.	Лек	2	ПК-1	Л1.1
	экзамен		27	ПК-1	
	Итого		108 час		

4.2 Заочная форма обучения (Всего 108 час., 14 час. аудиторных занятий).

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компе-тенции	УМИО
1	2	3	4	5	6
Курс 4, Семестр 7					
Модуль 1:Концепция функционального программирования. - 48(8 +40СР)					
1.1	История функционального программирования: – общее представление о функциональном программировании (**ФП**) и его применении; – классификация языков функционального программирования; – история создания и развития языка **Haskell.**	Лек.	2	ПК-1	Л1.1
1.2	Недостатки императивного стиля программирования (принцип последовательного исполнения, влияние на скорость работы центрального процессора).	СР	2	ПК-1	Л1.1, Л1.2,
1.3	Два способа проектирования вычислительной системы с параллельным вычислением, достоинства и недостатки.	СР	2	ПК-1	Л1.1
1.4	Математические основы **ФП - λ(лямда)** исчисление Алозно Черча: – основные понятия λ- исчисления; – место и роль различных редукций в λ-исчислении; – процесс преобразования формул в λ-исчислении; – нормальный порядок редукций;	СР	6	ПК-1	Л1.1

	– методы представления констант и функций в чистом λ-исчислении.				
1.5	**Рекурсия** в λ-исчислении. Приведение выражения к нормальной и слабой заголовочной нормальной форме в системе редукций λ-исчисления.	СР	4	ПК-1	Л1.1
1.6	Упражнения по λ-исчислению. λ-исчисление как универсальный язык программирования	ПЗ1	2	ПК-1	Л1.1
1.7	Функциональный стиль программирования: – особенности функционального стиля; – отличительные особенности функций языков программирования в ФП; – «чистые» и «нечистые» функции, отсутствие присваиваний; – использование ***рекурсии*** вместо циклов	СР	6	ПК-1	Л1.1
1.8	Представление функций в виде суперпозиций обращений к другим функциям в ФП.	СР	2	ПК-1	Л1.1
1.9	Решение задач на языке Java или Паскаль в императивном стиле и с использованием функционального стиля: – вычисление числа ***e*** в традиционном и функциональном стиле; – нахождение корня функции методом ***бисекции*** в традиционном и функциональном стиле; – вычисление суммы элементов массива в традиционном и функциональном стиле.	СР	4	ПК-1	Л1.1
1.10	Система программирования Haskell Platform: – установка системы программирования Haskell Platform; – запуск интерпретатора выражений на языке Haskell – GHCi (интерпретатор командной строки) или (в системе Windows) WinGHCi. GHC (Glasgow Haskell Compiler) – компилятор программ на языке; – получение и просмотр полного списка команд; – установление различных режимов работы интерпретатора; – проверка типа выражений, которые подлежат вычислению; – вычисление простых и сложных выражений;	Лаб.1	2	ПК-1	Л1.1

	– загрузка определения функций для их последующего вызова (применения)				
1.11	Подготовка, загрузка и исполнение простых программы на языке Haskell с помощью установленного интерпретатора.	СР	4	ПК-1	Л1.1
1.12	Haskell – строго типизированный язык. Система типов языка Haskell: – целые с двумя подтипами(**Integer, Int**); – вещественные с двумя подтипами (**Float, Double**); – логические Bool (**True, False**); – символьные Char (выделение апострофами). Идентификаторы встроенных или определённых программистом типов, классов, модулей и пакетов. Идентификаторы объектов (простых и сложных типов, функций). Применение апострофов для построения имён функций. Собственные идентификаторы.	Лек.	2	ПК-1	Л1.1
1.13	Элементарные типы данных. Полный список разрешенных операций для класса **Num.** Функция **fromInteger** для явного преобразования типов. Объединение типов в кортежи. Стандартные функции и операции. Две особенности записи выражений	СР	4	ПК-1	Л1.1
1.14	Определение функций с помощью уравнений: – задание типа функции и уравнения определения функции; – редукция–преобразование выражений. Осуществление вычисления выражений в Haskell (исполнение программы) с помощью последовательных редукций исходного выражения Приведение исходного выражения к ***нормальной форме.***	СР	6	ПК-1	Л1.1
	Модуль 2: Техника работы со списками, функции высших порядков. **Час** 60(6+54СР)				
2.1	Редукция выражений, работа которых не определяется уравнениями, содержащих примитивные встроенные операции (арифметические операции). ***Рекурсивно*е** определение функции. Вычисление условного выражения	СР	4	ПК-1	Л1.1
2.2	Определение функций с помощью уравнений; вычисление выражений в **Haskell** (исполнение	СР	6		Л1.1

	программы) с помощью последовательных редукций исходного выражения; определение простой рекурсивной функции.			ПК-1	
2.3	Навыки написания программ на языке Haskell; технологии обработки списковых структур; техника преобразования рекурсивных функций в функции с концевой рекурсией**.**	Лаб.2	2	ПК-1	Л1.1
2.4	***Концевая рекурсия и*** *накапливающие аргументы*. Нахождение чисел Фибоначчи с помощью простого рекурсивного определения. Эффективное вычисление чисел Фибоначчи. Приближенное вычисление числа ***e*** - пример функции с накапливающими аргументами	СР	4	ПК-1	Л1.1
2.5	Вычисление факториала числа с помощью ***концевой рекурсии***	СР	2	ПК-1	Л1.1
2.6	Техника работы со списками: Определение и обозначение типа списка. Задание объектов-списков перечислением. Два полезных сокращения для записи списков. Конструктор списков. Операции обработки списков ***(head, last, tail, !!, null, length,*** ++). Суммирование элементов списка. Обращение списка.	СР	4	ПК-1	Л1.1
2.7	Обработка списков строк, элементами, служат списки символов. Функция проверки наличия в списке «белых» строк	СР	2	ПК-1	Л1.1
2.8	Написать функцию (программу), определяющую количество строк в списке, содержащих буквы.	Ср	4	ПК-1	Л1.1
2.9	Написать функцию сортировки списка методом «простых вставок» с помощью заголовка модуля (**module** Sort (simleSort) **where**). Вариант функции *simleSort* и функции insert с концевой рекурсией. Сравнение результатов с точки зрения замедления работы	СР	4	ПК-1	Л1.1
2.10	Подготовка функций (программ): – Метод слияния. Функция сортировки списка слиянием (сортировка фон Неймана) – Два варианта функций, вычисляющих	СР	4	ПК-1	Л1.1

	число сочетаний из ***n*** предметов по ***m.*** Сравнение по быстродействию.				
2.11	Написать функцию (программу), вычисляющую длину самой длинной неубывающей последовательности подряд идущих чисел в заданном списке элементов	СР	4	ПК-1	Л1.1
2.12	Понятие *функций* ***высшего порядка***, область их применения в функциональных программах. (foldl, foldr) – ***операции свёртки списка***. Обработка списков с помощью функций высших порядков. Использование функций фильтрации (condition, filtered, quicksort). Использование функций высших порядков при обработке сложных структур	Лек	2	ПК-1	Л1.1
2.13	Использование функций высших порядков при обработке сложных структур данных. Функции отображения и свёртки.	СР	4	ПК-1	Л1.1
2.14	Преобразование карринговых функций в некарринговые и обратно. Функциональное представление данных. Представление множеств характеристическими функциями	СР	4	ПК-1	Л1.1
2.15	Навыки применения функций высших порядков в программировании на языке Haskell; навыки избавления от рекурсии при обработке сложных структур данных	СР	2	ПК-1	Л1.1
2.16	Класс отображаемых структур данных Functor. Моноид в языке Haskell (класс Monoid в пакете Data.Monoid). Значение и область применимости этих классов. Определение нейтрального элемента (***mempty***), бинарной операции (***mconcat***), соединение элементов списка моноидных значений с помощью операции (***mappend***).	СР	6	ПК-1	Л1.1
2.17	Ввод-вывод. Компиляция программ на Haskell.	ПЗ2	2	ПК-1	Л1.1

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Рекомендуемая литература				
5.1.1. Основная литература				
Код	**Авторы, составители**	**Заглавие**	**Издательство, год**	**Кол.**
Л1.1	Конева С. И.	Функциональное программирование: Учебное пособие, часть 1	Ростов-на-Дону СКФ МТУСИ, 2018	Э1
5.1.2 Дополнительная литература				
Код	**Авторы, составители**	**Заглавие**	**Издательство, год**	**Кол.**
Л2.1	Уорбэртон Р.	Лямда–выражения в Java 8	М.: ДМК Пресс, 2014.	
Л2.2	Душкин Р. В.	Функциональное программирование на языке Haskell	М.: ДМК Пресс, 2016.	
Л2.3	Кубенский А. А.	Функциональное программирование	М.: Юрайт, 2016	
Л2.4	Зыков С. В.	Программирование. Функциональный подход.: учебник и практикум для академического бакалавриата.	М.: Юрайт, 2016	
5.2 Электронные образовательные ресурсы				
Э1	http://www.skf-mtusi.ru/?page_id=659.			
Э2	http://www. haskell.org/platform/.			
5.3 Программное обеспечение				
П.1	MS Visio			
П.2	MS Word, MS Excel			
П.3	MS Power Point			

6. Материально - техническое обеспечение дисциплины

6.1 МТО лекционных занятий	
1	Лекционная аудитория, оснащенная проектором, ПК (ноутбуком), экраном
6.2 МТО практических занятий	
1	Компьютерные аудитории с возможностью выхода в локальную сеть Филиала и Интернет(аудитории: 218, 214, 202, 305)
6.3 МТО рубежных контролей и экзамена.	
1	Компьютерные аудитории с возможностью выхода в локальную сеть Филиала и Интернет (аудитории: 218, 214, 202, 305)

7. Методические рекомендации указания для обучающихся по самостоятельной работе

Самостоятельная работа студентов является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих курсов (систем), а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачётам и экзаменам.

Постановку задачи обучаемым на проведение самостоятельного занятия преподаватель осуществляет на одном из занятий, предшествующему данному. Он разъясняет смысл занятия и указывает, что к нему студенты должны приготовить. Задание на самостоятельную работу должно быть выдано заблаговременно с тем, чтобы студенты имели время на информационный поиск в библиотеке необходимых пособий.

Методику самостоятельной работы все обучаемые выбирают индивидуально.

На самостоятельную работу студентам дневной формы обучения выносится материал, представленный в таблице 3

Очная форма обучения

Таблица 3

№ п/п	Темы, разделы, вынесенные на самостоятельную подготовку, вопросы для подготовки к практическим и лабораторным занятиям; курсовые работы, содержание контрольных работ; рекомендации по использованию литературы, ЭВМ и др.	Часов всего 17	*Неделя*
1	***Модуль 1 Концепция ФП***		
	1.Недостатки императивного стиля программирования (принцип последовательного исполнения, влияние на скорость работы центрального процессора).	1	1
	2.Два способа проектирования вычислительной системы с параллельным вычислением, достоинства и недостатки.	1	2
	3.Рекурсия в λ-исчислении. Приведение выражения к нормальной и слабой заголовочной нормальной форме в системе редукций λ-исчисления	2	3
	4.Представление функций в виде суперпозиций обращений к другим функциям в ФП.	1	4-5
2	**Модуль 2 Функции высших порядков**		
	1.Вычисление факториала числа с помощью ***концевой рекурсии***	1	6-7
	2.Обработка списков строк, элементами которых служат списки символов.	1	8-10
	3.Написать функцию сортировки списка методом «простых вставок» с помощью заголовка модуля (**module** Sort (simleSort) **where**).	1	11-12
	4.Подготовка функций (программ): Метод слияния. Функция сортировки списка слиянием (сортировка фон Неймана	2	13
	5.Написать функцию (программу), вычисляющую длину самой длинной неубывающей последовательности подряд идущих чисел в заданном списке элементов	2	14-15
	6.Класс отображаемых структур данных Functor. Моноид в языке Haskell (класс Monoid в пакете Data.Monoid).	2	16-17
3	***Подготовка к экзамену***	***27***	

Студенты заочной формы обучения могут осваивать вопросы для самостоятельного изучения в удобное для них время.

Дополнения и изменения в рабочей программы