

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Сыров Игорь Анатольевич  
Должность: Директор  
Дата подписания: 21.08.2025 20:17:11  
Уникальный программный ключ:  
b683afe664d7e9f64175886cf9626a198149ad36

СТЕРЛИТАМАКСКИЙ ФИЛИАЛ  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Факультет *Математики и информационных технологий*  
Кафедра *Прикладной информатики и программирования*

**Аннотация рабочей программы дисциплины (модуля)**

дисциплина ***Б1.В.ДВ.02.01 Функциональное программирование***

часть, формируемая участниками образовательных отношений

Направление

***02.03.03 Математическое обеспечение и администрирование информационных систем***

код наименование направления

Программа

***Сетевое программирование и администрирование информационных систем***

Форма обучения

***Очная***

Для поступивших на обучение в  
***2020 г.***

Стерлитамак 2023

**1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в образовательной программе индикаторами достижения компетенций**

<b>Формируемая компетенция (с указанием кода)</b>	<b>Код и наименование индикатора достижения компетенции</b>	<b>Результаты обучения по дисциплине (модулю)</b>
ПК-3. Способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	ПК-3.1. Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений.	Обучающийся должен: знать особенностей функционального подхода в разработке программного обеспечения.
	ПК-3.2. Умеет программировать в рамках этих направлений.	Обучающийся должен: уметь использовать функциональные методы при решении простых вычислительных задач.
	ПК-3.3. Имеет практический опыт разработки программ в рамках этих направлений.	Обучающийся должен: владеть методами использования в профессиональной деятельности функционального языка программирования.

**2. Цели и место дисциплины (модуля) в структуре образовательной программы**

Цели изучения дисциплины:

Дисциплина изучается на 4 курсе в 7 семестре

**3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость (объем) дисциплины составляет 2 зач. ед., 72 акад. ч.

<b>Объем дисциплины</b>	<b>Всего часов</b>
	<b>Очная форма обучения</b>
Общая трудоемкость дисциплины	72
Учебных часов на контактную работу с преподавателем:	
лекций	16
практических (семинарских)	

лабораторных	16
другие формы контактной работы (ФКР)	0,2
Учебных часов на контроль (включая часы подготовки):	
зачет	
Учебных часов на самостоятельную работу обучающихся (СР)	39,8

<b>Формы контроля</b>	<b>Семестры</b>
зачет	7

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела / темы дисциплины	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)			
		Контактная работа с преподавателем			СР
		Лек	Пр/Сем	Лаб	
<b>1</b>	<b>Введение в функциональное программирование.</b>	<b>8</b>	<b>0</b>	<b>5</b>	<b>18</b>
1.1	Понятие функции и функциональной программы.	2	0	0	5
1.2	Строго функциональные языки.	2	0	0	5
1.3	Основы функционального программирования.	4	0	5	8
<b>2</b>	<b>Язык программирования Haskell.</b>	<b>8</b>	<b>0</b>	<b>11</b>	<b>21,8</b>
2.1	Типы данных языка Haskell.	2	0	4	5
2.2	Программирование на Haskell.	2	0	7	5
2.3	Представление и интерпретация функциональных программ.	2	0	0	6,8
2.4	Особенности интерпретирования императивных программ.	2	0	0	5
	<b>Итого</b>	<b>16</b>	<b>0</b>	<b>16</b>	<b>39,8</b>

##### 4.2. Содержание дисциплины, структурированное по разделам (темам)

Курс лабораторных занятий

№	Наименование раздела / темы дисциплины	Содержание
<b>1</b>	<b>Введение в функциональное программирование.</b>	
1.3	Основы функционального программирования.	Лабораторная работа №1 Тема: " Основы функционального программирования на языке Haskell". Цель работы: Изучить среду программирования, научиться создавать, отлаживать и исполнять программы в среде HUG98.

<b>2</b>	<b>Язык программирования Haskell.</b>	
2.1	Типы данных языка Haskell.	Лабораторная работа №2 Тема: "Типы данных языка Haskell". Цель работы: научиться применять типы языка программирования данных.
2.2	Программирование на Haskell.	Лабораторная работа №3 Тема: "Создание программ на Haskell". Цель работы: научиться разрабатывать программы на языке программирования Haskell. Лабораторная работа №4 Тема: "Обработка списков". Цель работы: научиться работать со списками (создавать новый список, сортировать).

#### Курс лекционных занятий

№	Наименование раздела / темы дисциплины	Содержание
<b>1</b>	<b>Введение в функциональное программирование.</b>	
1.1	Понятие функции и функциональной программы.	Понятие функции и функциональной программы. История развития языков функционального программирования. Программирование при помощи функций.
1.2	Строго функциональные языки.	Императивный, объектно-ориентированный, логический и функциональный подходы к программированию – достоинства, недостатки и основные характеристики. Общие сведения о функциональном подходе к программированию.
1.3	Основы функционального программирования.	Основы функционального программирования на языке Haskell. Основы языка Haskell. Символы, константы, атомы, логические значения. Базовые функции. Элементарные понятия; приемы программирования.
<b>2</b>	<b>Язык программирования Haskell.</b>	
2.1	Типы данных языка Haskell.	Символьные данные и S-выражения. Типизация в функциональных языках. Строгая типизация. Списки и деревья. Элементарные селекторы и конструкторы с примерами применения.
2.2	Программирование на Haskell.	Рекурсивные функции. Примеры применения рекурсивных функций на различных задачах обработки списков. Проблема выбора подфункций. Проблема модульности функциональной программы. Возможность накапливающих параметров на примере инверсии списка. Локальные определения в функциональных программах. Точечная запись выражений.
2.3	Представление и интерпретация функциональных программ.	Представление и интерпретация функциональных программ. Абстрактная и конкретная формы программ. Проблема связывания значений и переменных. Техника между функциональными и императивными программами.
2.4	Особенности	Особенности интерпретирования императивных

	интерпретирования императивных программ.	программ. Функциональные эквиваленты императивных программ. Аппаратное обеспечение функциональных программ.
--	--	---