


МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой  
Математического обеспечения ЭВМ  
наименование кафедры, отвечающей за реализацию дисциплины  
 Абрамов Г.В.  
подпись, расшифровка подписи  
23.03.2024 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**Б1.В.ДВ.03.02 Функциональное программирование**

*Код и наименование дисциплины в соответствии с учебным планом*

1. Код и наименование направления подготовки/специальности:

02.03.02 Фундаментальная информатика и информационные технологии

2. Профиль подготовки/специализация:

Инженерия программного обеспечения

3. Квалификация (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: МО ЭВМ

6. Составители программы: Авсеева О.В., к.т.н., доцент  
(ФИО, ученая степень, ученое звание)

7. Рекомендована: Научно-методическим советом факультета прикладной математики, информатики и механики, протокол №10 от 15.06.2019

*(наименование рекомендующей структуры, дата, номер протокола,*

Продлена научно-методическим советом факультета ПММ, протокол №7 от 26.05.2023  
*отметки о продлении вносятся вручную)*

Продлена НМС факультета ПММ протокол №5 от 22.03.2024 г.

8. Учебный год: 2027-2028

Семестр(ы): 7

9. Цели и задачи учебной дисциплины:

Целями освоения учебной дисциплины являются: освоение основ и методов функционального программирования и выработка практических навыков применения этих знаний.

Задачи учебной дисциплины: формирование у студентов представления об основных принципах функционального программирования, их основных применений в современном программировании.

**10. Место учебной дисциплины в структуре ООП:** учебная дисциплина относится к формируемой участниками образовательных отношений части Блока 1 и является дисциплиной по выбору. Предшествующими для данной дисциплины являются дисциплины Информатика и программирование, Дискретная математика.

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-6	Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии	ПК-6.1	Использует в профессиональной деятельности объектно-ориентированные системы программирования	знать основы $\lambda$ -исчисления, основы программирования на языке Haskell уметь: проектировать рекурсивные алгоритмы и реализовывать простые рекурсивные алгоритмы на языке Haskell
		ПК-6.3	Систематизирует материал, необходимый для разработки технической документации на программную продукцию.	уметь: разрабатывать программные приложения для решения поставленных задач на функциональном языке программирования владеть: навыками отладки функциональных программ

**12. Объем дисциплины в зачетных единицах/час.(в соответствии с учебным планом) —**  
3 / 108.

**Форма промежуточной аттестации(зачет/экзамен) зачет.**

**13. Трудоемкость по видам учебной работы**

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
		3 семестр	
Аудиторные занятия	64	64	
в том числе:	лекции	16	32
	практические	32	16
	лабораторные	16	32
Самостоятельная работа	44	44	
Форма промежуточной аттестации (зачет – 0 час. / экзамен – 36час.)	0	0	
Итого:	108	108	

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
<b>1. Лекции</b>			
1.1	Введение в функциональное программирование.	Введение в курс. Парадигма функционального программирования. Сравнение императивного и функционального программирования. Характерные черты функционального программирования.	
1.2	Основы лямбда-исчисления	<p>Причины использования формализации лямбда - исчисления. Понятие лямбда - выражения. Каррирование. Свободные и связанные переменные, выражения</p> <p>Лямбда-исчисление как формальная система. Типы - выражений. Подстановки. Конверсии. Равенство лямбда -термов. Экстенциональность. Редукция лямбда - термов. Редукционные стратегии. Комбинаторы.</p> <p>Представление данных в лямбда - выражении. Булевские значения и условия. Пары и кортежи. Натуральные числа. Рекурсивные функции. Именованные выражения. "Синтаксический сахар"</p> <p>Типы в лямбда - исчислении. Парадокс Рассела. Типизированное лямбда -исчисление. Базовые типы. Типизация по Черчу и Карри. Формальные правила типизации. Полиморфизм. Let- полиморфизм. Наиболее общие типы. Сильная нормализация. Понятие отложенные вычисления (ленивые). Применение отложенных вычислений. Примеры функций, использующих отложенные вычисления.</p> <p>Классы типов в лямбда - исчислении. Экземпляры классов. Методы классов. Расширения.</p> <p>Монады. Использование монад. Do-нотация. Встроенные монады. Монада типу Maybe. Монады состояний. Операции ввода-вывода.</p>	
<b>2. Практические занятия</b>			
2.1	Основы лямбда-исчисления	Упражнения по $\lambda$ -исчислению, $\lambda$ -исчисления как универсальный язык программирования.	
2.2	Язык Haskell	<p>Комментарии. Рекурсия. Операция выбора и правила выравнивания. Кусочное задание функций. Сопоставление с образцом. Построение списков.</p> <p>let-связывание. Сигнализация об ошибках. Охраняющие условия. Полиморфные типы. Пользовательские типы. Пары.</p> <p>Множественные конструкторы. Классы типов. Синонимы типов</p> <p>Определение операторов. Рекурсивные типы. Списки как рекурсивные типы. Синтаксические деревья</p> <p>Модули. Абстрактные типы данных. Операции</p>	

		ввода-вывода. Базовые операции ввода-вывода. Стандартные операции ввода-вывода. Создание исполняемых программ	
<b>3. Лабораторные занятия</b>			
3.1	Язык Haskell	Типы. Арифметика. Кортежи. Списки. Строки. Функции. Условные выражения. Функции многих переменных и порядок определения функций.	
		Комментарии. Рекурсия. Операция выбора и правила выравнивания. Кусочное задание функций. Сопоставление с образцом. Построение списков. Некоторые полезные функции	
		Функция map. Функция filter. Функции foldr и foldl. Другие функции высшего порядка. Лямбда-абстракции. Секции.	
		let-связывание. Сигнализация об ошибках. Охраняющие условия. Полиморфные типы. Пользовательские типы. Пары. Множественные конструкторы. Классы типов. Синонимы типов	
		Определение операторов. Рекурсивные типы. Списки как рекурсивные типы. Синтаксические деревья	
		Модули. Абстрактные типы данных. Операции ввода-вывода. Базовые операции ввода-вывода. Стандартные операции ввода-вывода. Создание исполняемых программ	

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в функциональное программирование.	2	-	-	2	4
2	Основы лямбда-исчисления	14	4	-	8	26
3	Язык Haskell	-	12	32	34	78
	Итого	16	16	32	44	108

### 14. Методические указания для обучающихся по освоению дисциплины

При изучении дисциплины рекомендуется использовать конспекты лекций, основную и дополнительную литературу, ресурсы интернет.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей по работе на LMS-платформе, своевременно подключаться к online-занятиям, соблюдать рекомендации по организации самостоятельной работы

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Хендерсон, Питер. Функциональное программирование : Применение и реализация / П. Хендерсон ; пер. с англ. Л.Т. Петровой; под ред. А.П. Ершова .— М. : Мир, 1983 .— 349,[1] с. : ил. — (Мат. обеспечение ЭВМ) .
2	Сергиевский, Георгий Максимович. Функциональное и логическое программирование : [учебное пособие для студ. вузов, обуч. по направлению "Информатика и вычисл. техника"] / Г.М. Сергиевский, Н.Г. Волченков .— М. : Академия, 2010 .— 317, [1] с. : ил., табл. — (Высшее профессиональное образование. Информатика и вычислительная техника) .— Библиогр.: с.313-314 .— ISBN 978-5-7695-6433-8.

б) дополнительная литература:

№ п/п	Источник
4	GRAHAM HUTTON Programming in Haskell Second Edition University Printing House, Cambridge CB2 8BS, United Kingdom

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
1.	www.lib.vsu.ru
2.	www.ohaskell.guide – Шевченко Д., О Haskell по-человечески

**16. Перечень учебно-методического обеспечения для самостоятельной работы** (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник
1.	Иванов И.П., Сборник задач по курсу "Алгоритмы и структуры данных": Метод. указания / И.П. Иванов, А. Ю. Голубков, С. Ю. Скоробогатов. - М. : Издательство МГТУ им. Н. Э. Баумана, 2013. - 32 с. - ISBN 978-5-7038-3681-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <a href="https://www.studentlibrary.ru/book/ISBN9785703836811.html">https://www.studentlibrary.ru/book/ISBN9785703836811.html</a> . - Режим доступа : по подписке.
2.	Представление, хранение и обработка информации в примерах, задачах и упражнениях : Проб. учеб. пособие / О.Ф.Ускова, Н.В. Огаркова, И.Е. Воронина и др.; Воронеж. гос. ун-т. — Воронеж, 2002. — 176 с. : ил. — ISBN 5-9273-0266-1 : 57.00.

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение:**

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий. Для организации занятий рекомендован online-курс «Объектно-ориентированный анализ и проектирование», размещенный на платформе Электронного университета ВГУ (LMS Moodle), а также интернет-ресурсы, приведенные в п. 15в

**18. Материально-техническое обеспечение дисциплины:**

Терминальная рабочая станция SunRay 2 (терминалы с мониторами) (15 шт.)

Мультимедиа-проектор Nec NP100

Коммутатор HP ProCurve Switch 1400-24G

Компьютер Intel Core i5-2400 CPU@3.10 GHz, ОЗУ 8Гб, 500Гб HDD

Моноблок Apple iMac MD093RU/A Core i5 (2.70)8 Гб/1 Тб/GeForce GT640M 512Мб/21,5" (15 шт.)

Компьютер APPLE Mac Pro MD772RU/A Xeon W3565/8 гб,2 Тб

Коммутатор HP ProCurve Switch 1400-24G

Мультимедиа-проектор BENQ MX503 DLP 2700 лм, 13000:1, 1024\*768

Терминальная рабочая станция SunRay2 (терминалы с мониторами) (15 шт.)

Коммутатор D-Link DES-1016D

**19. Оценочные средства для проведения текущей и промежуточной аттестаций**

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Введение в функциональное программирование.	ПК-6	ПК-6.3	Устный опрос
2.	Основы лямбда-исчисления	ПК-6	ПК-6.3	Устный опрос Практическое задание
3.	Язык Haskell	ПК-6	ПК-6.3	Лабораторные работы

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
Промежуточная аттестация форма контроля - зачет				<i>Перечень вопросов Практическое задание</i>

## 20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

### 20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

*Практикоориентированные задания  
Лабораторные работы*

#### Примеры практических заданий

1. На плоскости заданы  $n$  точек своими координатами  $(x_1, y_1), (x_2, y_2)$ . Составить программу вычисления максимального внутреннего и минимального внешнего радиусов кольца с центром в начале координат, содержащего все точки. Координаты точек считывать из файла.
2. Составить программу, которая по натуральному числу  $N$  и выдает все трехзначные числа, сумма цифр которых равна  $N$ .
3. Программа должна считывать с командной строки имена двух файлов и выводить на экран те строки этих файлов, которые отличаются друг от друга.
4. Программа, принимающая имя файла и выводящая его строки, перед каждой из которых записывается ее номер.
5. Программа считывает из файла матрицу и выводит на экран суммы столбцов этой матрицы.

#### Примеры заданий для лабораторных работ

1. Составить программу, которая по натуральному числу  $N$  и основанию системы счисления  $p$  выводит цифры  $A_k$  представления  $N$  в  $p$ -ичной системе счисления.
2. Программа считывает из файла матрицу и выводит на экран след этой матрицы (сумму диагональных элементов).
3. Программа считывает две матрицы из файлов и записывает в третий файл матрицу, являющуюся их суммой.
4. Программа считывает из одного файла матрицу, а из другого – вектор и записывает в третий файл результат умножения матрицы на вектор.

### 20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторные работы

Зачет проводится в виде устного собеседования по выполненным лабораторным работам.

### Описание критериев и шкалы оценивания результатов обучения при промежуточной аттестации

Для оценивания результатов обучения используются следующие показатели:

- 1) умение реализовывать различные алгоритмы на языке Haskell
- 3) владение приемами функционального программирования

Для оценивания результатов обучения используется шкала: «зачтено», «не зачтено». Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности	Шкала оценок
---------------------------------	--------------------------	--------------

	КОМПЕТЕНЦИЙ	
<i>Полное соответствие ответа обучающегося всем перечисленным критериям. Продемонстрировано умение реализовывать различные алгоритмы на языке Haskell, знание основ лямбда исчисления</i>	<i>Повышенный уровень</i>	<i>Зачтено</i>
<i>Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки, не умеет решать задачи с применением АСД</i>	–	<i>Не зачтено</i>