

МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра математического моделирования

УТВЕРЖДАЮ

Декан математического факультета



Нестеров П.Н.

20 июня 2023 г.

Рабочая программа дисциплины
Функциональное программирование

Направление подготовки (специальности)
01.03.02 Прикладная математика и информатика

Направленность (профиль)
«Прикладное программирование и информационные технологии»

Форма обучения очная

Программа рассмотрена
на заседании кафедры
от 11 апреля 2023 г., протокол № 8

Программа одобрена НМК
математического факультета
протокол № 9 от 3 мая 2023 г.

1. Цели освоения дисциплины

Целями освоения дисциплины «Функциональное программирование» являются:

1. знакомство с функциональным подходом программирования;
2. знакомство с рядом сопутствующих технологий: виды типов (обобщения старших порядков), сопоставление с образцом;
3. умение решать задачи общего программирования в функциональном ключе;
4. умение работать с монадами и функторами;

2. Место дисциплины в структуре образовательной программы

1. Дисциплина «Функциональное программирование» является дисциплиной по выбору и относится к вариативной части Блока 1.
2. Для успешного изучения дисциплины «Функциональное программирование» необходимы знания и умения, полученные в результате освоения дисциплины «языки программирования и методы трансляции». Освоение дисциплины может явиться хорошим подспорьем для изучения математических моделей, так как функциональные языки наиболее приближены к математической нотации; дает навыки, которые помогают в освоении языка специализированных математических программ.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК-3 Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	ИД-ПК-3.1 Обладает устойчивыми знаниями в области разработки алгоритмов и программирования	Знать: <ul style="list-style-type: none">• отличительные особенности функционального программирования;• основные функторы и монады, а также области их применения;• способы создания и работы с типами данных и синтаксическими конструкциями на языке программирования Haskell
	ИД-ПК-3.2 Имеет навыки разработки и реализации алгоритмов в области системного и прикладного	Знать: <ul style="list-style-type: none">• отличительные особенности функционального программирования;• основные функторы и монады, а также области их применения;

	программного обеспечения	<ul style="list-style-type: none"> • способы создания и работы с типами данных и синтаксическими конструкциями на языке программирования Haskell <p>Уметь:</p> <ul style="list-style-type: none"> • решать типовые задачи, используя функциональный подход, использовать рекурсию, отложенные вычисления, свёртки, функторы и монады <p>Владеть:</p> <ul style="list-style-type: none"> • навыками разработки программного обеспечения на языке программирования Haskell
	ИД-ПК-3.3 Обладает способностью критического анализа и совершенствования разрабатываемых алгоритмов и программ	<p>Знать:</p> <ul style="list-style-type: none"> • отличительные особенности функционального программирования; • основные функторы и монады, а также области их применения; • способы создания и работы с типами данных и синтаксическими конструкциями на языке программирования Haskell <p>Уметь:</p> <ul style="list-style-type: none"> • решать типовые задачи, используя функциональный подход, использовать рекурсию, отложенные вычисления, свёртки, функторы и монады; • видеть недостатки и оптимизировать код; <p>Владеть:</p> <ul style="list-style-type: none"> • навыками разработки программного обеспечения на языке программирования Haskell • навыками критического анализа и совершенствования разрабатываемых алгоритмов и программ

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	С е м ес тр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						сам осто ятел ьная рабо та	Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа							
			лек ции	пр акт ич еск ие	лаб ора тор ны е	ко нс уль тац ии	атте ста цио нны е исп ыта ния			
1.	Типы данных, классы типов, суперклассы: - Встроенные типы данных и создание своих типов данных; - Классы типов Eq, Ord, Show, Read, Enum, Bounded, Num, Floating, Integral.	7	2	2	4	1			6	Задания для самостоятельной работы; Лабораторная работа 1.
2.	Рекурсивные типы данных: - Интервалы, кортежи, списки, и генераторы списков.; - Рекурсивные функции replicate, take, reverse, repeat, zip, elem.	7	2	2		1			7	Задания для самостоятельной работы
3.	Декларативный и композиционный стиль: - Синтаксис функций; - Синтаксические конструкции; - Сопоставление с образцом.	7	2	2					6	Задания для самостоятельной работы
4.	Функции высшего порядка: - Каррированные функции; - Функции zipWith, flip, map, filter; - Свертки; - Комбинатор неподвижной точки.	7	2	2	6	1			5	Задания для самостоятельной работы; Лабораторная работа 2.
5.	Приоритет инфиксных операций: - Приоритет операций;	7	1	1					3	Задания для самостоятельной работы

	- Создание операций; - Функции композиции и применения.								
6.	Функторы и монады: - Классы Functor, Applicative и Monad; - Монада State - Тип данных Maybe;	7	2	2		1		6	Задания для самостоятельной работы
7.	Ввод-вывод. Взаимодействие с консолью: - Компилирование программы; - Методы getContents, getLine, putStr, putStrLn, getChar, print; - Функции return, when, sequence, mapM, forever, forM	7	1	1				6	Задания для самостоятельной работы
8.	Ввод-вывод. Работа с файлами: - Чтение и запись файлов; - ByteString.	7	2	2	6	1		7	Задания для самостоятельной работы; Лабораторная работа 3
9.	Исключения.	7	1	1		1		2	Задания для самостоятельной работы
10.	Ленивые чудеса: - Дифференцирование; - Интегрирование; - Степенные ряды.	7	1	1				6	Задания для самостоятельной работы
						2	0,5	33,5	Экзамен
	ИТОГО 144 часа		16	16	16	8	0,5	87,5	

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

- Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития отрасли и практики, достижения в этой сфере, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы. Обзор рынка труда.
- Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная

информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

- Практическое занятие – занятие, посвященное практической отработке у студентов умений и навыков применения базовых знаний при разработке приложений или его компонентов, по приобретению умений и навыков самостоятельного решения и анализа задач, связанных с разработкой приложений. На практических занятиях студенты решают поставленные перед ними задачи под руководством (контролем) преподавателя. Обсуждение процесса решения задачи и оценка правильности полученного результата (постановки задачи, выбора метода ее решения, проверка полученного результата и т.д.) в ходе практического занятия проводится коллективно студентами под руководством преподавателя.
- Лабораторное занятие – групповое занятия, направленные на творческое усвоение теоретических основ учебной дисциплины. Данные занятия способствуют приобретению умений и навыков самостоятельной постановки и реализации задания в рамках тематики с учетом требований преподавателя.
- Консультации – групповые занятия, на которых по просьбе студентов рассматриваются наиболее сложные моменты теории и практических задач.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются:

- Glasgow Haskell Compiler
- Notepad++
- Adobe Acrobat Reader DC
- Internet Explorer и/или иные браузеры

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

- Автоматизированная библиотечная информационная система "БУКИ-NEXT" http://www.lib.uniyl.ac.ru/opac/bk_cat_find.php
- Электронно-библиотечная система «Юрайт» <https://www.biblio-online.ru/>
- Электронно-библиотечная система «Лань» <http://e.lanbook.com/>
- НЭБ Национальная электронная библиотека <https://rusneb.ru/>

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература

1. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / А. А. Кубенский. — Москва : Издательство Юрайт, 2020. — 348 с. — (Высшее

образование). — ISBN 978-5-9916-9242-7 <https://urait.ru/bcode/451097> (электронный ресурс)

б) дополнительная литература

1. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2021. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. <https://urait.ru/bcode/470387> (электронный ресурс)

в) ресурсы сети «Интернет»

Документация и справочники:

1. Официальный сайт [Haskell Language](#)
2. [Documentation](#)
3. [Hackage: Introduction](#)

Онлайн компилятор:

1. [Online Haskell Compiler - online editor](#)

Статьи:

1. [Функциональное программирование — Википедия](#)
2. [Функциональное программирование для всех / Хабр](#)
3. [WebDev → Императивное vs Функциональное программирование](#)
4. [Functional programming vs. imperative programming - LINQ to XML](#)
5. [Haskell — Википедия](#)

Пособия:

1. [Мягкое введение в Haskell](#)

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа и практических занятий
- (семинаров);
- учебные аудитории для проведения занятий лабораторных работ, лаборатория
- информационных технологий;
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств
- обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в

электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы):

Ст. преподаватель каф. математического моделирования А.М. Ковалева

**Приложение № 1 к рабочей программе дисциплины
«Функциональное программирование»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости**

Примеры заданий для самостоятельной работы студентов:

Тема №1.

Реализуйте свои типы данных даты, дней недели и времени

Тема №2.

Найдите прямоугольный треугольник (длины сторон с точностью до перестановок), удовлетворяющий следующим условиям: длины сторон являются целыми числами; длина каждой стороны меньше либо равна 10; периметр треугольника (то есть сумма длин сторон) равен 24.

Тема №3.

Определите модуль, который будет вычислять площади простых фигур, треугольника, окружности, прямоугольника, трапеции. Помните, что фигуры могут задаваться различными способами.

Тема №4.

Реализуйте метод нахождения чисел фибоначчи, находящихся в заданном диапазоне.

Тема №5.

Создать свои операции сложения и умножения, чтобы они давали $2+2*2 = 8$.

Тема №6.

У нас есть два правила размножения клеток-букв в организме. На каждом этапе мы во всём слове заменяем букву a на слово ab и букву b на a. Начав с одной буквы a, мы за несколько шагов пришли к более сложному слову. Реализуйте данный процесс.

Тема №7.

Написать консольную программу ввода-вывода для нахождения корней квадратного трехчлена.

Тема №8.

Реализовать приложение, которое позволяет работать в списке дел (хранится в файле). Пользователь должен иметь возможность: просмотреть пронумерованный список дел, удалить задачу по номеру, добавить задачу в заданную позицию.

Тема №9.

Доработайте программу из двух предыдущих тем, добавив обработку исключений.

Тема №10.

Реализовать метод Пикара для решения краевой задачи.

Пример задания для лабораторной работы №1:

Создать модуль для работы с комплексными числами:

- Модуль должен содержать минимальное количество импортов (только то, что необходимо).
- Вывод на печать комплексного числа должен производиться в том виде, в котором мы привыкли его видеть.
- С комплексными числами производить операции сложения, вычитания и сравнения, а также получение модуля.
- Необходимо использовать классы типов
- Операции должны обозначаться привычным образом (+) и (-), (==) и (/=)
- Используйте классы Eq и Show

Примеры заданий для лабораторной работы №2:

1. Удалить каждый N-й элемент из списка.
2. Определите, являются ли два положительных целых числа взаимно простыми.
3. Узнайте, является ли список палиндромом.
4. Если список содержит повторяющиеся элементы, их следует заменить одной копией элемента. Порядок элементов не должен быть изменен.
5. Упакуйте последовательные дубликаты элементов списка в подписки. Если список содержит повторяющиеся элементы, они должны быть помещены в отдельные подписки.
`λ> pack ['a', 'a', 'a', 'a', 'b', 'c', 'c', 'a', 'a', 'd']`
`["aaaa", "b", "cc", "aa", "d"]`
6. Расшифруйте закодированный список длин серий.
`λ> decodeModified [(4,'a'),(1,'b'),(2,'c'),(2,'a'),(1,'d'),(4,'e')]`
`"aaaabccaadeeee"`
7. Разделить список на две части; длина первой части дана.
`λ> split "abcdefghik" 3`
`("abc", "defghik")`
8. Удалить K-й элемент из списка.
`λ> removeAt 2 "abcd"`
`("b", "acd")`
9. Вставьте элемент в заданную позицию в списке.
`λ> insertAt 'X' "abcd" 2`
`"aXbcd"`

Примеры заданий для лабораторной работы №3:

1. Вывести результат табулирования функции $y = x^2$ в файл. Интервал табулирования и шаг должен вводить пользователь.
2. В файле следует оставить слова, которые начинаются и заканчиваются на одинаковую букву.
3. В файле следует оставить только строки с четным номером.
4. Запросить у пользователя имя файла и число. Оставить в файле слова содержащие указанное число символов (остальные - удалить).
5. В файле находится последовательность целых чисел (одно число на одной строке). Нужно вывести на консоль, является ли последовательность возрастающей.

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Список теоретических вопросов на экзамене:

1. Типы данных.
2. Классы типов.
3. Декларативный и композиционный стили. Синтаксические конструкции.
4. Списки как типы. Генераторы списков.
5. Функции работы со списками. Свертки.
6. Приоритет инфиксных операций.
7. Функции высшего порядка.
8. Ввод-вывод. Консоль.
9. Чтение и запись файлов.
10. Bytestring.
11. Исключения.
12. Функторы и монады.
13. Особенности функционального подхода в программировании.

**Приложение № 2 к рабочей программе дисциплины
«Функциональное программирование»**

Методические указания для студентов по освоению дисциплины

Дисциплина «Функциональное программирование» проводится в форме практических и лабораторных занятий, направленных на закрепление базовых знаний, полученных на лекционных занятиях этого курса. Перед экзаменом предусмотрена консультация по разбору наиболее трудных моментов. Для успешного освоения дисциплины очень важна активная работа, как в аудитории, так и самостоятельно при выполнении домашних заданий.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Функциональное программирование» самостоятельно студенту крайне сложно. Это связано со сложностью изучаемого материала и достаточно большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым.