

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ  
Декан ЭФ

УТВЕРЖДАЮ /Н.М. Стрельникова/  
(Ф.И.О. декана (директора института))

11.03.2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

М.1.2.6 Функциональное программирование

*(код и наименование дисциплины по учебному плану)*

Направление подготовки  
(специальность)

09.04.03 Прикладная информатика

Квалификация выпускника

Магистр

*(бакалавр/магистр/специалист)*

Программа магистратуры

Искусственный интеллект в бизнес-аналитике

Курс 2  
Семестр 4

**Распределение учебного времени**

Трудоемкость по учебному плану	144 / 4	часов/зачетных единиц
Лекции	20	часов
Лабораторные работы	20	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	40	часов
Контактная работа по экзамену	-	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	104	часов
Самостоятельная работа по подготовке к экзамену	-	часов
Экзамен	-	семестр
Зачет	-	семестр
БРК, ДЗ	4	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.04.03 Прикладная информатика

Программу составили:

заведующий кафедрой с ученой степенью кандидата наук	ИСЭ	СОГЛАСОВАНО	Т.А. Уразаева
(должность)	(кафедра)		(И.О. Фамилия)
	ИиСП	СОГЛАСОВАНО	А.В. Бородин
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина  
Кафедра информационных систем в экономике

	(наименование кафедры)		
13.02.2024	протокол №	6	
(дата)			
Заведующий кафедрой	СОГЛАСОВАНО	Т.А. Уразаева	
		(И.О. Фамилия)	

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)  
кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Т.А. Уразаева
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит  
выпускающая кафедра

СОГЛАСОВАНО	О.Е. Иванов
	(И.О. Фамилия)

Эксперт(ы): Усков Юрий Викторович, генеральный директор ООО «Ричмедиа»

Рабочая программа проверена и зарегистрирована в УМЦ 12.03.2024 г.

Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

## Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ПК-11 Способен разрабатывать и модернизировать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности в различных предметных областях	ПК-11.1. Разрабатывает программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности а различных предметных областях	<b>знания:</b> ПК – 11.1 З-1. Знает новые научные принципы и методы разработки программного и аппаратного обеспечения технологий и систем искусственного интеллекта для решения профессиональных задач в различных предметных областях. <b>умения:</b> ПК – 11.1 У-1. Умеет разрабатывать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности для решения профессиональных задач в различных предметных областях. <b>навыки:</b> ПК – 11.1 В-1. Владеет навыками использования базовых концепций функционального программирования.
	ПК-11.2. Модернизирует программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности а различных предметных областях	<b>знания:</b> ПК – 11.2 З-1. Знает особенности модернизации программного и аппаратного обеспечения технологий и систем искусственного интеллекта для решения профессиональных задач в различных предметных областях. <b>умения:</b> ПК – 11.2 У-1. Умеет модернизировать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности для решения профессиональных задач в различных предметных областях. <b>навыки:</b> ПК – 11.2 В-1. Владеет навыками использования инструментальных средств функционального программирования при решении профессиональных задач.

## Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к элективным дисциплинам (модулям) ОПОП.

Дисциплина является элективной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Проектирование интеллектуальных информационных систем (ПК-11)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих практиках: Преддипломная практика (ПК-11); государственной итоговой аттестации в форме: Выполнение и защита выпускной квалификационной работы (ПК-11)

### Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: дискуссионные, исследовательские, лекционные занятия, практические и лабораторные занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, информационные, классическая лекция, проблемная лекция

### Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### 4 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
<b>Основные концепции функционального программирования</b>	<b>72</b>	ПК-11
Лекция. Лекция №1. Знакомство с задачами, приведшими к возникновению концепции ФП	4	
Лекция. Лекция №2. Язык программирования LISP. Символьные вычисления	2	
Лекция. Лекция №3. Современное состояние ФП. Языки Erlang, Ruby, F#	4	
Лабораторная работа. Лабораторная работа №1. Исследование задач, решение которых хорошо укладывается в концепцию функционального программирования.	4	
Лабораторная работа. Лабораторная работа №2. Основные концепции программирования на языке LISP.	6	
Задания для самостоятельной работы, в том числе выполнение КР Знакомство с задачами, приведшими к возникновению концепции ФП.  Язык программирования LISP. Символьные вычисления.  Современное состояние ФП. Языки Erlang, Ruby, F#.	52	
<b>Язык функционального программирования Haskell и его применение</b>	<b>72</b>	ПК-11
Лекция. Лекция №4. ЯФП Haskell. Знакомство с интерпретатором ЯФП Haskell	4	
Лекция. Лекция №5. Haskell: синтаксис, каррирование, теоретико-множественная конструкция функции и ее отражение в ЯП Haskell, сопоставление с образцом, списки	4	
Лекция. Лекция №6 Методы функционального программирования, используемые в технологиях искусственного интеллекта	2	
Лабораторная работа. Лабораторная работа №3. Знакомство с интерпретатором ЯФП Haskell	4	
Лабораторная работа. Лабораторная работа №4. Решение задачи по вариантам с использованием ЯФП Haskell	6	

Задания для самостоятельной работы, в том числе выполнение КР		
Разбор примеров использования ЯФП Haskell. Примеры ленивых вычислений в ЯФП Haskell. ФП в традиционных ЯП. Символьные вычисления.		
Метапрограммирование на ЯФП.		
ЯФП в системах поддержки принятия решений.	52	
Иная контактная работа:	0	

## Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности. **Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **занятиям семинарского типа** включает ознакомление с планом лабораторного занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Изучение дисциплины включает выполнение контрольной работы, лабораторной работы. хода освоения дисциплин представлены в рабочей программе. Условия аттестации приведены в технологической карте, входящей в состав рабочей программы дисциплины. Формой промежуточной аттестации по дисциплине является балльно-рейтинговый контроль.

## Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
<b>УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ</b>		
1.	Душкин, Роман Викторович. Функциональное программирование на языке Haskell [Текст] / Р. В. Душкин. Москва: ДМК Пресс, 2007. - 605 с. ISBN 5-94074-335-8. Экземпляры: всего 43.	43

2.	Душкин, Роман Викторович. Функциональное программирование на языке Haskell (+CD) [Электронный ресурс] / Р. В. Душкин. Москва: ДМК Пресс, 2008. - 608 с. ISBN 5-94074-335-8.	<a href="http://e.lanbook.com/books/element.php?pl1_cid=25&amp;pl1_id=1247">http://e.lanbook.com/books/element.php?pl1_cid=25&amp;pl1_id=1247</a>
3.	Сергиевский, Георгий Максимович. Функциональное и логическое программирование [Текст] : [учеб. пособие для вузов по направлению "Информатика и вычисл. техника"] / Г. М. Сергиевский, Н. Г. Волченков. М.: Академия, 2010. - 317, [1] с. ISBN 978-5-7695-6433-8. Экземпляры: всего 6.	6

## 6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	314а (III)	ПК 1 - ICL RAY P222.1 ,клавиат.,мышь.,монитор NEC 23" LCD EX 231WP-BK (1), ПК 2 - ICL RAY P222.2 ,клавиат.,мышь.,монитор NEC 23" LCD EX 231WP-BK (15), Проектор мультимедийный Hitachi CP-RX93 (1), Шкаф IBM Netbay 25U (1), Экран настенный рулонный 200x200 см (1), Комплект учебной мебели (1)	Microsoft Windows Enterprise, Microsoft Office Standard, Агент Dr.Web, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач, 1С:Документооборот 8 КОРП , 1С:Предприятие 8. Комплект для обучения.

## Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает	удовлетворительно

	затруднения в выполнении практических заданий.	
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

#### 7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

#### 7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

1. Стандартизированный чистый функциональный язык программирования общего назначения:

[+] а) Haskell

[-] б) Scala

[-] в) Ruby

2. Haskell является одним из самых распространённых языков программирования с поддержкой таких вычислений:

[-] а) дополнительных

[+] б) отложенных

[-] в) второстепенных

3. Поскольку Haskell язык функциональный, то основная управляющая структура – это:

☐ а) матрица

☐ б) таблица

☒ в) функция

4. Haskell принадлежит к семейству языков:

☒ а) ML

☐ б) SL

☐ в) CL

5. Непосредственно на Haskell оказал большое влияние язык:

☐ а) Мюррея

☐ б) Овранта

☒ в) Миранда

6. В этом году была предложена первая версия языка, Haskell 0:

☐ а) 1995

☒ б) 1990

☐ в) 1998

7. Одна из основных характеристик языка Haskell:

☐ а) всегда полное применение

☐ б) дополнительное применение

☒ в) частичное применение

8. Одна из основных характеристик языка Haskell:

☒ а) ленивые вычисления

☐ б) резкие вычисления

☐ в) мгновенные вычисления

9. Одна из основных характеристик языка Haskell:



☐ а) не сопоставление с образцом

☐ б) мгновенные вычисления

☒ в) сопоставление с образцом

10. Быстрый компилятор, поддерживающий стандарт Haskell 98, написанный на Haskell 98; доступен для всех юникс-подобных платформ, включая MacOS X и Windows/cygwin:

☐ а) HBI и HBC

☒ б) nhc98

☐ в) YHC

11. Форк nhc98, ставящий целью быть более переносимым и эффективным, поддерживает отладчик Nat; генерирует промежуточный байт-код, который можно использовать для генерации кода на других языках программирования:

☒ а) YHC (York Haskell Compiler)

☐ б) Yale Haskell

☐ в) UHC

12. Интерпретатор и компилятор, поддерживающие стандарт Haskell 98 с некоторыми расширениями; точно работает в Linux, поддерживается слабо:

☐ а) LHC

☒ б) HBI и HBC

☐ в) UHC

13. Альтернативный кодогенератор для GHC, поддерживающий глобальную низкоуровневую оптимизацию программы в целом:

☒ а) LHC (The Luxurious LHC Haskell Optimization System)

☐ б) jhc

☐ в) Yale Haskell

14. Экспериментальный компилятор для исследования возможностей глобальной оптимизации программ:

☐ а) YHC

☐ б) LHC

☒ в) jhc

15. Ранняя реализация языка Haskell на Lisp:

[+] а) Yale Haskell

[-] б) nhc98

[-] в) YHC

16. Рекомендованный базовый дистрибутив для разработчиков:

[-] а) LLVM

[+] б) Haskell Platform

[-] в) YHC

17. Мозаичный оконный менеджер Xmonad для X Window System целиком написан на языке:

[-] а) Dylan

[-] б) Ruby

[+] в) Haskell

18. Распределённая система управления версиями с рядом уникальных возможностей – написана на Haskell:

[-] а) Darts

[+] б) Darcs

[-] в) Durts

19. Строго типизированный интерфейс доступа к SQL-базам данных:

[-] а) HDBC

[-] б) Takusen

[+] в) haskeIldb

20. Универсальный интерфейс доступа к реляционным базам данных (поддерживаются SQLite, MySQL, PostgreSQL, ODBC):

[-] а) Takusen

[+] б) HDBC

[-] в) haskeIldb

21. Библиотека доступа к базам данных, использующая интерфейс левой свёртки:

☒ a) Takusen

☐ б) haskeIldb

☐ в) HDBC

22. Редактор диаграмм:

☐ a) Diagrams

☒ б) Blobs

☐ в) FieldTrip

23. Библиотека для программирования схем и диаграмм:

☐ a) FieldTrip

☐ б) GPipe

☒ в) Diagrams

24. Библиотека для функционального описания 3D-объектов:

☐ a) HaskellCharts

☒ б) FieldTrip

☐ в) SDL

25. Функциональный интерфейс для программирования GPU:

☐ a) GLUT

☐ б) Yampa

☒ в) GPipe

26. Библиотека для создания графических интерфейсов на основе GTK+:

☐ a) Grapefruit и wxFruit

☒ б) Gtk2Hs

☐ в) qtHaskell

27. Библиотека для создания графических интерфейсов на основе wxWidgets:

[+] а) wxHaskell

[-] б) Gtk2Hs

[-] в) HTk

28. Вики-платформа, основанная на системах контроля версиями (darcs, git или mercurial):

[-] а) happstack

[-] б) Twidge

[+] в) giti

29. Многофункциональная библиотека для работы с XML:

[+] а) HaXml

[-] б) HXT

[-] в) Pandoc

30. Инструмент для сетевой установки, автоматической сборки и развёртывания приложений на Haskell:

[-] а) happy

[+] б) cabal-install

[-] в) haddock

Перечень вопросов для проведения промежуточной аттестации

1. Сравнение императивного и декларативного подходов к программированию. Примеры языков.
2. Функциональное программирование. Основные особенности.
3. История ФП. Возникновение. Семейства языков.
4. Чистые функции. Особенности программирования на чистых функциях. Функции высших порядков. Каррирование.
5. Ленивая и энергичная (строгая) стратегии исполнения. Разновидности и примеры.
6. Язык Haskell. Синтаксис объявления и определения функций и переменных. Встроенные выражения. Префиксная и инфиксная запись.
7. Язык Haskell. Приоритет и ассоциативность операций. Каррирование и секции.
8. Язык Haskell. Встроенные типы данных.
9. Язык Haskell. Пользовательские типы данных.
10. Язык Haskell. Pattern matching.
11. Функциональные комбинаторы. Назначение и примеры. ФК над списками и функциями.

12. Функциональные комбинаторы. Назначение и примеры. Свёртки.
13. Классы типов в Haskell. Определение и примеры.
14. Классы типов в Haskell. Моноиды и функторы
15. Классы типов в Haskell. Монады.
16. Монады Reader, Writer, State.
17. Монады-трансформеры.
18. Хвостовая рекурсия.
19. Линзы в Haskell. Основные определения.
20. Линзы в Haskell. Композиция линз. Законы линз.
21. Traversal в Haskell. Связь с линзами. Законы.
22. Ленивое выполнение. Особенности ленивых алгоритмов.
23. Ленивое выполнение. Работа со списками в ленивом языке.
24. Ленивое выполнение. Потoki.
25. Ленивое выполнение. Сильносвязные структуры данных: двусвязные списки и деревья.