

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО
Руководитель ОПОП

_____ М.В. Коломина

«8» сентября 2022 г.

УТВЕРЖДАЮ
Зав. кафедрой ПМИ

_____ М.В. Коломина

«8» сентября 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Функциональное программирование»

Составители

**Корнеев Г.А., к.т.н., доцент ФИТиП, ИТМО
Маврин П.Ю., ассистент ФИТиП, ИТМО**

Направление подготовки / специ-
альность

**01.03.02 Прикладная математика и
информатика**

Направленность (профиль) ОПОП

**Программирование и искусственный
интеллект**

Квалификация (степень)

бакалавр

Форма обучения

очная

Год приёма

2023

Курс

3

Семестр(ы)

5

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целями освоения дисциплины «Функциональное программирование» является изучение и практическое освоение средств функционального программирования для решения научных и прикладных задач.

1.2. Задачи освоения дисциплины:

- изучение классических способов решения задач с использованием функционального программирования.
- формирование практических навыков разработки архитектурных решений для реализации распределенной высокопроизводительной системы с использованием функционального программирования.
- формирование умения отлаживать распределенные решения, реализованные на функциональном языке программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина «Функциональное программирование» относится к части, формируемой участниками образовательных отношений и осваивается в 5 семестре.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения, навыки, формируемые предшествующими учебными дисциплинами (модулями):

Архитектура компьютера

Язык программирования C++

Технологии программирования

Знания: базовые конструкции построения алгоритмов, представление целочисленных и вещественных данных на компьютере.

Умения: использовать основные конструкции языков программирования, выбирать типы данных, соответствующие решаемой задаче.

Навыки: программирования на C++, решения типовых задач.

2.3. Последующие учебные дисциплины и (или) практики, для которых необходимы знания, умения, навыки, формируемые данной учебной дисциплиной:

- Проектирование программного обучения,
- Написание выпускной квалификационной работы,
- Производственная практика.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс освоения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки / специальности:

а) профессиональных (ПК).

- ПК-11. Разработка и сопровождение программных проектов.
- ПК-14. Способность определять эффективный способ решения прикладных задач с применением информационных технологий и программной инженерии, разрабатывать и внедрять соответствующие программные решения.

Таблица 1 – Декомпозиция результатов обучения

Код компетенции	Планируемые результаты освоения дисциплины (модуля)		
	Знать (1)	Уметь (2)	Владеть (3)
ПК-11 ПК-11.1. Формализация и алгоритмизация поставленных задач ПК-11.2. Проектирование программного обеспечения	ИПК-11.1.1 Классические способы решения задач с использованием функционального программирования.	ИПК-11.2.1 Разрабатывать архитектурные решения для реализации распределенной высокопроизводительной системы с использованием функционального программирования.	ИПК-11.3.1 Навыками отладки распределенных решений, реализованных на функциональном языке программирования.
ПК-14 ПК-14.1. Способность проектировать и реализовывать программные решения с применением методов функционального, автоматного и эволюционного программирования ПК-14.2. Способность проектировать и реализовывать мобильные и web-приложения	ИПК-14.1.1 Представления программ расширенного лямбда-исчисления.	ИПК-14.2.1 Работать со строками на функциональном языке.	ИПК-14.3.1 Навыками применения функций высшего порядка на языке Haskell.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Объем дисциплины составляет 3 зачетные единицы, в том числе 68 часов, выделенных на контактную работу обучающихся с преподавателем (из них 34 часа – лекции, 34 часа – лабораторные работы) и 40 часов – на самостоятельную работу обучающихся.

Таблица 2 – Структура и содержание дисциплины

№ п/п	Наименование раздела, темы	Семестр	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости Форма промежуточной аттестации
			Л	ПЗ	ЛР	КР	СР	
1	Понятие о функциональном программировании	5	6		7		8	Лабораторная работа №1
2	Введение в язык Haskell	5	6		7		8	Лабораторная работа №2
3	Даты, классы, экземпляры	5	7		7		8	Лабораторная работа №3
4	Моноиды	5	8		5		8	Лабораторная работа №4
5	Интерпретация и компиляция функциональных программ	5	9		7		8	Лабораторная работа №5, 6
ИТОГО			34		34		40	Экзамен

Таблица 3. Матрица соотнесения разделов, тем учебной дисциплины и формируемых в них компетенций

Разделы, Темы дисциплины (модуля)	Кол-во часов	Компетенции		
		ПК-11	ПК-14	Общее количество компетенций
Раздел 1. Понятие о функциональном программировании	22,75	+	+	2
Раздел 2. Введение в язык Haskell	21	+	+	2
Раздел 3. Даты, классы, экземпляры	21	+	+	2
Раздел 4. Моноиды	21	+	+	2
Раздел 5. Интерпретация и компиляция функциональных программ	21	+	+	2
Консультации	1			
Контроль промежуточной аттестации	0,25			
Итого	108			

Краткое содержание каждой темы дисциплины (модуля)

№ раздела	Наименование раздела дисциплины	Содержание
-----------	---------------------------------	------------

1	Понятие о функциональном программировании	Почему функциональное программирование в Haskell, Приближенное значение
2	Введение в язык Haskell	Строки, Базовый синтаксис
3	Даты, классы, экземпляры	Даты, классы, экземпляры, Строки 2
4	Моноиды	Моноиды, Дерево
5	Интерпретация и компиляция функциональных программ	Лямбда-исчисление, Интерпретация, Граф, Представление функциональных программ

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Лекционные занятия

Основной формой реализации теоретического обучения является лекция, которая представляет собой систематическое, последовательное изложение преподавателем-лектором учебного материала теоретического характера. Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Порядок подготовки лекционного занятия включает в себя выполнение следующих этапов:

- изучение требований программы дисциплины;
- определение целей и задач лекции;
- разработка плана проведения лекции;
- подбор литературы (ознакомление с методической литературой, публикациями периодической печати по теме лекционного занятия);
- отбор необходимого и достаточного по содержанию учебного материала;
- определение методов, приемов и средств поддержания интереса, внимания, стимулирования творческого мышления студентов;
- написание конспекта лекции.

Лекция должна включать следующие разделы:

- формулировку темы лекции;
- указание основных изучаемых разделов или вопросов и предполагаемых затрат времени на их изложение;
- изложение вводной части;
- изложение основной части лекции;
- краткие выводы по каждому из вопросов;
- заключение;
- рекомендации литературных источников по излагаемым вопросам.

Лабораторные занятия

Лабораторное занятие – целенаправленная форма организации педагогического процесса, направленная на углубление научно-теоретических знаний и овладение определенными методами работы, в процессе которых вырабатываются умения и навыки выполнения тех или иных учебных действий в данной сфере науки. Они развивают научное мышление и речь, позволяют проверить знания студентов и выступают как средства оперативной обратной связи.

Правильно организованные лабораторные занятия ориентированы на решение следующих задач:

- обобщение, систематизация, углубление, закрепление полученных на лекциях и в процессе самостоятельной работы теоретических знаний по дисциплине (предмету);

- формирование практических умений и навыков, необходимых в будущей профессиональной деятельности, реализация единства интеллектуальной и практической деятельности;
- выработка при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Состав заданий для лабораторного занятия должен быть спланирован с расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством учащихся.

Лабораторные занятия должны так быть организованы, чтобы студенты ощущали нарастание сложности выполнения заданий, испытывали бы положительные эмоции от переживания собственного успеха в учении, поисками правильных и точных решений.

Самостоятельная работа

Самостоятельная работа – это вид учебной деятельности, которую студент совершает в установленное время и в установленном объеме индивидуально или в группе, без непосредственной помощи преподавателя (но при его контроле), руководствуясь сформированными ранее представлениями о порядке и правильности выполнения действий.

В учебном процессе образовательного учреждения выделяются два вида самостоятельной работы:

- аудиторная – выполняется на учебных занятиях, под непосредственным руководством преподавателя и по его заданию (выполнение самостоятельных работ; выполнение контрольных и практических работ; решение задач);
- внеаудиторная – выполняется по заданию преподавателя, но без его непосредственного участия (подготовка к аудиторным занятиям; изучение учебного материала, вынесенного на самостоятельную проработку; выполнение домашних заданий разнообразного характера; выполнение индивидуальных заданий, направленных на развитие у студентов самостоятельности и инициативы; подготовка к контрольной работе). Внеаудиторные самостоятельные работы представляют собой логическое продолжение аудиторных занятий, проводятся по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Лекция

- Лекция – основной вид обучения в вузе.
- В лекции излагаются основные положения теории, ее понятия и законы, приводятся факты, показывающие связь теории с практикой.
- Накануне лекции необходимо повторить содержание предыдущей лекции (а также теорию по изучаемой теме в школьных учебниках геометрии, если эта тема была представлена в них), а затем посмотреть тему очередной лекции по программе (по плану лекций).
- Полезно вести записи (конспекты) лекций: для непонятных вопросов оставлять место при работе над темой лекции с учебными пособиями.
- Записи лекций следует вести в отдельной тетради, оставляя место для дополнений во время самостоятельной работы.
- При конспектировании лекций выделяйте главы и разделы, параграфы, подчеркивайте основное.

Лабораторное занятие

- Лабораторное занятие – наиболее активный вид учебных занятий в вузе. Он предполагает самостоятельную работу над лекциями и учебными пособиями.
- К каждому лабораторному занятию нужно готовиться. Подготовку следует начинать с повторения теории (по записям лекций или по учебному пособию). После этого нужно решать задачи из предложенного домашнего задания.

Организация самостоятельной работы

Самостоятельность в учебной работе способствует развитию заинтересованности студента в изучаемом материале, вырабатывает у него умение и потребность самостоятельно получать знания, что весьма важно для специалиста с высшим образованием. Самостоятельная работа студентов представлена в следующих формах:

- работа с учебной литературой и конспектом лекций с целью подготовки к лабораторным занятиям, составление конспектов тем, выносимых на самостоятельную проработку;
- систематическое выполнение домашних работ.

Таблица 4 – Содержание самостоятельной работы обучающихся

Темы/вопросы, выносимые на самостоятельное изучение	Кол-во часов	Формы работы
Раздел 1. Понятие о функциональном программировании	8	Подготовка отчета по лабораторной работе №1.
Раздел 2. Введение в язык Haskell	8	Подготовка отчета по лабораторной работе №2.
Раздел 3. Даты, классы, экземпляры	8	Подготовка отчета по лабораторной работе №3.
Раздел 4. Моноиды	8	Подготовка отчета по лабораторной работе №4.
Раздел 5. Интерпретация и компиляция функциональных программ	8	

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля), выполняемые обучающимися самостоятельно

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.

Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.

Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине «Функциональное программирование» могут использоваться электронное обучение и дистанционные образовательные технологии.

6.1. Образовательные технологии

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line или off-line в формах.

Таблица 5 – Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Понятие о функциональном программировании	Обзорная лекция	Не предусмотрено	Выполнение лабораторной работы
Введение в язык Haskell	Обзорная лекция	Не предусмотрено	Выполнение лабораторной работы
Даты, классы, экземпляры	Обзорная лекция	Не предусмотрено	Выполнение лабораторной работы
Моноиды	Обзорная лекция	Не предусмотрено	Выполнение лабораторной работы
Интерпретация и компиляция функциональных программ	Обзорная лекция	Не предусмотрено	Выполнение лабораторной работы

6.2. Информационные технологии

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- система управления обучением LMS Moodle;
- использование возможностей Интернета в учебном процессе (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление обучающихся с оценками и т.д.);
- использование электронных учебников и различных сайтов (например, электронные библиотеки, журналы и т.д.) как источник информации;
- использование возможностей электронной почты;
- использование средств представления учебной информации (электронных учебных пособий, применение новых технологий для проведения занятий с использованием презентаций и т.д.);
- использование интерактивных средств взаимодействия участников образовательного процесса (технологии дистанционного или открытого обучения в глобальной сети);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т.е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс).

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

Наименование программного обеспечения	Назначение
Adobe Reader	Программа для просмотра электронных документов
Платформа дистанционного обучения LMS Moodle	Виртуальная обучающая среда
Microsoft Office 2013, Microsoft Office Project 2013, Microsoft Office Visio 2013	Пакет офисных программ
7-zip	Архиватор
Microsoft Windows 7 Professional	Операционная система
Kaspersky Endpoint Security	Средство антивирусной защиты
Google Chrome	Браузер
OpenOffice	Пакет офисных программ

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронная библиотека «Астраханский государственный университет» собственной генерации на платформе ЭБС «Электронный Читальный зал – БиблиоТех». <https://biblio.asu.edu.ru>
2. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». www.studentlibrary.ru.
3. Электронная библиотечная система издательства ЮРАЙТ, раздел «Легендарные книги». www.biblio-online.ru
4. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем». <https://library.asu.edu.ru>

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине «Функциональное программирование» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последователь-

ным освоением дисциплин и прохождением практик, а в процессе освоения дисциплины – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6 – Соответствие разделов, тем дисциплины, результатов обучения по дисциплине и оценочных средств

№ п/п	Контролируемые разделы, темы дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Понятие о функциональном программировании	ПК-11, ПК-14	Лабораторная работа №1
2	Введение в язык Haskell	ПК-11, ПК-14	Лабораторная работа №2
3	Даты, классы, экземпляры	ПК-11, ПК-14	Лабораторная работа №3
4	Моноиды	ПК-11, ПК-14	Лабораторная работа №4
5	Интерпретация и компиляция функциональных программ	ПК-11, ПК-14	Лабораторная работа №5, 6

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Таблица 7 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 8 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание по подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задания

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторные работы

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.

Объем выполненной работы: каждая лабораторная работа содержит 3-5 задач.

Срок сдачи работы: лабораторные работы должны быть сданы в период прочтения курса. Сдача работы представляет собой предоставление отчёта в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.

Лабораторные работы

№ п/п	Номер раздела дисциплины	Наименование лабораторной работы	Трудоемкость, часов
1	1	Приближенное значение	8
2	2	Строки	8
3	3	Строки 2	4
4	4	Дерево	4
5	5	Граф	4
6	5	Лямбда-исчисление	4

Примеры заданий к лабораторной работе «Приближенное значение»

1. Студенты должны Написать программу для вычисления приближенного значения числа e по формуле для разложения e^x в ряд Тейлора. Предложить программы на языке Паскаль, написанные в традиционном императивном и функциональном стилях.
2. Студенты должны понимать особенности функционального стиля программирования
3. Студенты должны уметь писать программы в функциональном стиле на традиционных языках программирования

Примеры заданий к лабораторной работе «Строки»

1. Написать функцию, определяющую количество строк в списке, содержащих хотя бы одну букву (буквой будем называть символ, для которого функция `\texttt{isAlpha :: Char -> Bool}` выдает значение `True`).
2. Студенты должны знать основные конструкции языка Haskell
3. Студенты должны уметь писать программы на Haskell небольшого размера

Примеры заданий к лабораторной работе «Строки 2»

1. Написать функцию, вычисляющую длину самой длинной строки в заданном списке строк.
2. Студенты должны быть знакомы с функциями высших порядков
3. Студенты должны уметь реализовывать функции, принимающие другие функции в качестве аргумента
4. Студенты должны уметь применять на практике функции высших порядков, такие как `свертка` и `map`

Примеры заданий к лабораторной работе «Дерево»

1. Дерево задано с помощью следующего описания структуры данных. `data Tree a = Node a [Tree a]`
2. то есть дерево представляет собой корневой узел, содержащий некоторое значение произвольного типа `a` и список поддеревьев. Написать функцию, вычисляющую высоту дерева.
3. Студенты должны понимать основы лямбда-исчисления
4. Студенты должны уметь строить несложные интерфейсы
5. Студенты должны уметь писать программы, состоящие из нескольких модулей, связанных интерфейсами

Примеры заданий к лабораторной работе «Граф»

1. Структура графа задана списками смежности номеров вершин, то есть списком, элементами которого являются пары, состоящие из номера вершины и списка вершин, инцидентных ей. type Graph = [(Int, [Int])]
2. Написать функцию, которая проверяет, существует ли в графе путь, соединяющий вершины с двумя заданными номерами.

Примеры заданий к лабораторной работе «Лямбда-исчисление»

1. Выполнить редукцию выражения. В нормальном и аппликативном порядке редукций. В обоих случаях найти нормальную форму (НФ) и слабую заголовочную нормальную форму (СЗНФ) выражения.
2. Студенты должны понимать принципы компиляции и интерпретации программ
3. Студенты должны уметь выполнять редукцию лямбда-выражений
4. Студенты должны уметь реализовывать структуры данных на функциональных языках программирования, в рамках которого студенты отвечают на теоретические вопросы, проектируют и реализуют программы с применением навыков и умений, полученных в процессе освоения дисциплины.

Шаблон отчета по лабораторной работе

Отчет по лабораторной работе № _____

«Название лабораторной работы»

1. Цель и задачи лабораторной работы: _____
2. Методика проведения исследования: _____
3. Анализ погрешностей: _____
4. Результаты: _____
5. Выводы: _____

Требования к выполнению лабораторной работы

Отчеты по лабораторным работам должны быть отправлены на электронную почту преподавателя не позднее, чем через две недели после выдачи задания. Полученные выводы и графический материал должны быть информативными и корректными.

Перечень вопросов и заданий, выносимых на экзамен

Формат проведения экзамена – устный опрос в формате ответов на вопросы билета. При необходимости экзаменатор имеет возможность задать 2-3 дополнительных вопроса по темам лабораторных работ и лекционному материалу.

Перечень вопросов:

1. Why FP and Haskell
2. Basic Syntax
3. Datas, Classes, Instances
4. Kinda lazy monoidal types
5. Monads, part 1
6. Functors, Applicative Functors
7. Monads, part 2
8. RealWorld
9. Monad Transformers
10. Speeding up Haskell
11. Template Haskell and Lens
12. Parallel and Concurrent Haskell
13. Comonads
14. Enterprise Haskell
15. Advanced type features
16. Idris
17. Укажите преимущества статической типизации (минимум \$4\$).
18. Укажите преимущества чистоты (минимум \$4\$).

19. Укажите преимущества иммутабельности (минимум \$4\$).
20. В чем отличие `\texttt{data}` от `\texttt{newtype}`?
21. Что такое "<Currying (каррирование)>" и функции высшего порядка?
22. Что дает слово "<deriving>"? Что это в языке Haskell?
23. Как обновить `$b$` (скажем на `$2$`) в `\texttt{data A = A { b :: Int }; c = A { b = 1 }}`?
24. Приведите пример типа с `kind`'ом `\texttt{(* -> Constraint) -> Constraint}`.
25. Укажите `kind` следующего типа данных: `\texttt{data A f g = B (f g Int)}`.
26. Укажите `kind` следующего типа данных: `\texttt{data A f g a = B (f g) (g a)}`.
27. Укажите `kind` следующего типа данных: `\texttt{data A f g = B (f g) (g f)}`.
28. Укажите `kind` типа `\texttt{type C m a = (Monad m, Show a)}`.
29. Укажите `kind` типа `\texttt{type C p a = (p [a], Show a)}`.
30. Напишите `type class` `\texttt{Traversable}`.
31. Реализуйте `\texttt{sequence}` через `\texttt{traverse}`.
32. Реализуйте `\texttt{traverse}` через `\texttt{sequence}`.
33. Напишите реализацию `\texttt{Foldable}` для `\texttt{Maybe}`.
34. Напишите реализацию `\texttt{Foldable}` для списка.
35. Напишите реализацию `\texttt{Traversable}` для `\texttt{Maybe}`.
36. Напишите реализацию `\texttt{Traversable}` для `\texttt{Either}`.
37. Напишите реализацию `\texttt{Traversable}` для списка.
38. Напишите о Language Extensions, приведите примеры.
39. Напишите, что делают эти расширения языка: `\texttt{TypeSynonyms}`, `\texttt{MultiParamTypeClasses}`, `\texttt{ViewPatterns}`, `\texttt{RecordsWildCards}`.
40. Напишите функцию `\texttt{(<$) :: Functor f => a -> f b -> f a}`. Она должна работать следующим образом: `\texttt{5 <$ Just 0 == Just 5, 5 <$ Nothing == Nothing}`.
41. Напишите функцию `\texttt{(>$) :: Functor f => f a -> b -> f b}`. Она должна работать следующим образом: `\texttt{Just 0 >$ 5 == Just 5, Nothing >$ 5 == Nothing}`.
42. Напишите законы функтора.
43. Напишите класс типов `\texttt{Bifunctor}` и реализуйте его для `\texttt{Either}`.
44. Напишите класс типов `\texttt{Bifunctor}` и реализуйте его для пары.
45. Реализуйте функцию `\texttt{(<<$>>) :: (Functor f, Functor g) => (a -> b) -> f (g a) -> f (g b)}`.
46. Напишите `type class` `\texttt{Applicative}` и его реализацию для `\texttt{ZipList}`.
47. Напишите `type class` `\texttt{Applicative}` и его реализацию для `\texttt{((->) r)}`.
48. Есть функция `\texttt{g :: a -> b}` и объект `\texttt{x :: Applicative f => f a}`. Напишите два разных способа получить объект `\texttt{y :: Applicative f => f b}` из `$x$` с использованием `$g$`.
49. Реализуйте `\texttt{join}` через `\texttt{bind}`.
50. Напишите тип `\texttt{(>=>)}` и смысл этого оператора.
51. Покажите, синтаксическим сахаром для чего является `\texttt{do}`-нотация.
52. Что такое `\texttt{IO}`? Как теоретически это реализовано?
53. Отличие `\texttt{unsafePerformIO}` от `\texttt{unsafeInterleaveIO}`?
54. Напишите тип `\texttt{MaybeT}` и реализуйте его инстанс `\texttt{Monad}`.
55. Нарисуйте табличку отличий обычных типов и их трансформеров для известных вам трансформеров.
56. Напишите тип `\texttt{StateT}` и то, как определен `\texttt{State}` через `\texttt{StateT}`.
57. Укажите, что делает `\texttt{deepseq}` и как.
58. В чем разница между `\texttt{seq}` и `\texttt{deepseq}`?
59. Что такое `irrefutable patterns` и зачем они нужны?
60. Что такое `\texttt{BangPatterns}`? Когда их нужно использовать?
61. В чем разница между `\texttt{seq}` и `\texttt{BangPatterns}`?
62. Что такое `Deforestation`?
63. Что такое `Stream Fusion` и зачем он нужен?

64. Напишите, что значит тип `\texttt{ST}` и напишите основные функции по работе с ним.
65. Что такое `\texttt{STRef}` и в чём отличие от `IORef`?
66. Чем плохо использовать `\texttt{IORef}` и `\texttt{IOArray}`? Зачем нужны `\texttt{STRef}` и `\texttt{STArray}`?
67. В чём разница между `\texttt{[] []}` и `\texttt{\$()}`?
68. Как можно посмотреть AST-дерево для выражения?
69. Что такое `\texttt{Q}` в типах функций Template Haskell?
70. Напишите реализацию `\texttt{over}`.
71. Укажите операторные обозначения функций `\texttt{view}`, `\texttt{set}` и `\texttt{over}`. Есть ли отличие в типах функций и их операторных выражений?
72. Реализуйте `\texttt{over}` через `\texttt{view}` и `\texttt{set}`.
73. Реализуйте `\texttt{set}` через `\texttt{over}`.
74. Напишите функцию `\texttt{lens}`, которая принимает геттер и сеттер и возвращает линзу.
75. Чем линзы отличаются от призм?
76. Что такое изоморфизм (`\texttt{ISO}`)?
77. Напишите тип `\texttt{Iso}`.
78. Напишите тип функции `\texttt{iso}`.
79. Напишите тип функции `\texttt{from}` для `\texttt{Iso}`.
80. Что такое `\texttt{Strategy}`? Перечислите несколько стратегий и реализуйте некоторые. Зачем они нужны?
81. Что делает `\texttt{forkIO}`? Чем он отличается от `\texttt{forkFinally}`?
82. Опишите, что такое `\texttt{MVar}`, зачем он может быть нужен и несколько функций по работе с этим объектом.
83. Как в Haskell обстоят дела с DeadLock'ами?
84. Что такое `\texttt{STM}` (коротко), что позволяет делать и какие есть функции по работе с ним?
85. В чём отличие Haskell потоков от, например, потоков в Java?
86. Что такое RTS?
87. Укажите несколько полезных опций RTS.
88. Напишите ранг следующих функций и их запись типа с `\texttt{forall: String -> Maybe Int, Either a b -> Either c d, [a] -> (forall b . [b] -> b)}`.
89. Напишите, как иметь список объектов разных функторов, внутри каждого из которых значения одинакового типа, чтобы иметь возможность применить функции из этого значения в другое.
90. Зачем нужно расширение `\texttt{ExistentialQuantification}`?
91. В чём разница между `\texttt{-XRank2Types}` и `\texttt{-XRankNTypes}`. Зачем нужны оба?
92. Зачем нужно расширение языка `\texttt{-XExplicitForall}`?
93. Зачем нужно расширение языка `\texttt{-XScopedTypeVariables}` и как оно взаимодействует с `\texttt{forall}`?
94. Зачем нужно расширение языка `\texttt{-XTypeApplications}`?
95. Что такое typed holes и зачем они нужны?
96. Для чего нужно расширение языка `\texttt{-XPartialSignatures}`?
97. Можно ли создать следующий тип данных в Haskell: `\texttt{data a :~> b = (a -> b) :~> (b -> a)}`.
98. Что такое Functional Dependencies? Назовите какой-нибудь известный вам type class, в котором присутствуют функциональные зависимости.
99. Реализуйте `\texttt{instance Comonad}` для обычного `\texttt{Zipper}`.
100. `\texttt{IO}` использует абстракцию монад, какой аналог есть в мире комонад?
101. Как можно было бы сделать `\texttt{codo}` нотацию для комонад? И что бы происходило в этом синтаксическом сахаре?
102. Напишите, какие комонады двойственны монадам `\texttt{Reader}`, `\texttt{Writer}`, `\texttt{State}`.

103. Напишите, какие монады двойственны комонадам `\texttt{Traced}`, `\texttt{Store}`, `\texttt{Env}`.
104. Напишите комонаду `\texttt{Traced}` и инстанс `\texttt{Comonad}` для неё.
105. Напишите комонаду `\texttt{Env}` и инстанс `\texttt{Comonad}` для неё.
106. Напишите комонаду `\texttt{Store}` и инстанс `\texttt{Comonad}` для неё.
107. Напишите комонаду `\texttt{Stream}` и инстанс `\texttt{Comonad}` для неё.
108. Напишите класс `\texttt{ComonadTrans}`.
109. Что такое "`<\texttt{_|_}-eliminator>`"? Зачем это надо?
110. Что такое "`<тотальность>`" и какие преимущества она даёт?
111. Что такое `!`-идиома в Idris?
112. Что такое `\texttt{[| |]}`-идиома в Idris?
113. Реализуйте функцию `\texttt{take}` для вектора на Idris.
114. Напишите тип "`<зависимая пара>`" на Idris.
115. Реализуйте функцию `\texttt{filter}` для вектора на Idris.
116. Реализуйте функцию `\texttt{head}` для списка на Idris, которая компилируется только с гарантированно непустыми списками.

Порядок формирования экзаменационного билета:

Билеты состоят из 4-х вопросов:

- 1 вопрос – с 1 по 29 вопрос из перечня вопросов к экзамену;
- 2 вопрос – с 30 по 58 вопрос из перечня вопросов к экзамену;
- 3 вопрос – с 59 по 87 вопрос из перечня вопросов к экзамену;
- 4 вопрос – с 88 по 116 вопрос из перечня вопросов к экзамену.

Пример экзаменационного билета № 1

1. Вопрос «Basic Syntax»
2. Вопрос «Напишите тип `\texttt{StateT}` и то, как определён `\texttt{State}` через `\texttt{StateT}`»
3. Вопрос «Что такое RTS?»
4. Вопрос «Реализуйте функцию `\texttt{head}` для списка на Idris, которая компилируется только с гарантированно непустыми списками.»

Таблица 9. Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
ПК-11. Разработка и сопровождение программных проектов.				
1.	Задание закрытого типа	Выберите верный ответ. Что возвращает функция <code>Sys.getpid()</code>? 1. Количество доступных для запуска потоков 2. Количество запущенных параллельно потоков 3. Номер версии RStudio 4. Версию языка R 5. Идентификатор процесса R сессии	5	1-3
2.		Выберите верный ответ. Как определяются значения логического типа при формальном построении языка функционального программирования? 1. <code>true = 1</code>; <code>false = 0</code> 2. <code>true = λx.λy.x</code>; <code>false = λx.λy.y</code> 3. <code>true = λx.λy.xy</code>; <code>false = λx.λy.yx</code> 4. как атомарные константы T и F	2	1-3
3.		Выберите верный ответ.	2	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		Как представляется число n в виде нумерала Черча? 1. $n \rightarrow \lambda f.\lambda x.f^n x$ 2. $n \rightarrow \lambda f.\lambda x.f^n x$ 3. как список из n элементов 4. как список из $n+1$ -го пустого списка		
4.		Выберите верный ответ. Как определяется конструкция <code>let</code> ? 1. <code>let x=e1 in e2 => (λx.e2)e1</code> 2. <code>let x=e1 in e2 => (λx.e1)e2</code> 3. <code>let x=e1 in e2 => (λf.[x/f]e2)e1</code> 4. <code>Y(λf.[x/f]e1) e2</code>	1	1-3
5.		Выберите верный ответ. Как определяется конструкция <code>letrec</code> ? 1. <code>letrec x=e1 in e2 => (λx.e2)e1</code> 2. <code>letrec x=e1 in e2 => (λx.e1)e2</code> 3. <code>letrec x=e1 in e2 => (λf.[x/f]e1)e2</code> 4. с помощью комбинатора неподвижной точки Y	4	1-3
6.	Задание открытого типа	Что такое замыкание?	Функция с некоторым контекстом вычислений, сохраненным на момент определения функции.	3-5
7.		Пусть геометрическое преобразование определяется функцией трансляции координат $\text{int}^* \text{int} \rightarrow \text{int}^* \text{int}$. Мы хотим определить функцию сдвига <code>translate : int*int</code> , которая возвращалась бы замыкание. Как это сделать?	<code>let translate (dx,dy) = fun (x,y) → (x+dx, y+dy)</code>	3-5
8.		Что такое мемоизация?	Запоминание результатов вычисления функции как при ленивых, так и при энергичных вычислениях	3-5
9.		Какой принцип построения функциональных программ?	Программа строится из набора функций, каждая из которых перерабатывает входные данные в выходные. Функции также могут рассматриваться как данные	3-5
10.		В чем отличия функционального программирования и императивного?	Функциональное программирование оперирует функциями и их применением к данным, императивное – операторами и тем, как они изменяют состояние памяти	3-5
11.	Задание комбинированного типа	Верно ли утверждение: Функциональные программы не содержат побочных эффектов, поскольку запрещено модифицировать внутренние переменные функции извне самой функции. Ответ обоснуйте.	Утверждение неверно, так как функциональные программы не содержат побочных эффектов, поскольку отсутствует понятие переменной и оператора присваивания	3-5

ПК-14. Способность определять эффективный способ решения прикладных задач с применением информационных технологий и программной инженерии, разрабатывать и внедрять соответствующие программные решения.

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
12.	Задание закрытого типа	<i>Выберите верный ответ.</i> Можно ли в F# изменять контекст вычисления функции внутри замыкания? 1. нет 2. да, если внутри создаются локальные переменные, описанные как mutable 3. да, если внутри создаются ссылочные объекты ref 4. да, если при компиляции включена опция позднего связывания	3	1-3
13.		<i>Выберите верные ответы.</i> Возможно ли в F# использовать ленивые вычисления? 1. нет, F# - энергичный язык 2. да, используя конструкции Lazy/Force 3. да, используя последовательности seq 4. да, используя ключи компилятора	2, 3	1-3
14.		<i>Выберите верный ответ.</i> Пусть L – генератор последовательности длины n. Какова сложность операции map f L? 1. O(1) 2. O(n) 3. O(log n) 4. O(n ²)	1	1-3
15.		<i>Выберите верный ответ.</i> За счет чего функциональные программы содержат меньше ошибок? 1. функциональные программы не содержат побочных эффектов 2. функциональные программы короче 3. на функциональных языках автоматически контролируются ошибки типа переполнения буфера 4. функциональные программы более просты и понятны для программиста	1	1-3
16.	Задание открытого типа	<i>Выберите верный ответ.</i> Какие операторы традиционно отсутствуют в функциональных языках? 1. оператор присваивания 2. оператор присваивания и операторы циклов с пред- и пост-условиями 3. оператор присваивания и условный оператор 4. все операторы присутствуют	2	1-3
17.		Правила для обновления Haskell гласят...	1. все имена берутся из одного типа данных; 2. все имена определяют один конструктор; 3. имена упоминаются только один раз; 4. значение должно содер-	3-5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			жать все указанные имена.	
18.		Каким выражением можно заменить выражение <code>if e1 then e2 else e3</code> ?	<code>case e1 of True → e2 False → e3</code>	3-5
19.		Для чего используются модули в Haskell?	Для управления пространствами имен и для создания абстрактных типов данных	3-5
20.		Функция <code>readHex</code> в Haskell выполняет...	Считывает число без знака в шестнадцатеричной записи	3-5
21.		Как интерпретируется символ <code>></code> , если он является первым символом в строке?	Это часть программы при использовании грамотного программирования.	3-5

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю)

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	<i>Выполнение лабораторных работ</i>	6/6,5	40	Сроки указаны в Moodle
Всего			40	-
Блок бонусов				
2.	<i>Посещение занятий</i>		10	
Всего			10	-
Дополнительный блок**				
3.	<i>Экзамен</i>			
Всего			50	-
ИТОГО			100	-

Таблица 11 – Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине

Сумма баллов	Оценка по 4-балльной шкале
90–100	5 (отлично)
85–89	4 (хорошо)
75–84	
70–74	
65–69	3 (удовлетворительно)
60–64	
Ниже 60	2 (неудовлетворительно)

При реализации дисциплины в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation (3rd Edition). — Addison-Wesley, Boston, MA, USA, 2006. — 750 с.
2. Шень А. Программирование: теоремы и задачи. — М.: МЦНМО, 2014. — 296 с.
3. Шень А., Верещагин Н. Языки и исчисления. — М.: МЦНМО, 2012. — 240 с.

4. Верещагин, Н. К. Колмогоровская сложность и алгоритмическая случайность [Электронный ресурс] / Н. К. Верещагин, В. А. Успенский, А. Шень. — Электрон. дан. — СПб: Лань, 2013. — 575 с. — Режим доступа: <https://e.lanbook.com/book/56395> — Загл. с экрана.
5. Городняя, Л. В. Парадигма программирования : учебное пособие для вузов - 2-е изд., стер. / Л. В. Городняя. Санкт-Петербург : Лань, 2021. - 232с. - Текст: электронный. - URL: [e.lanbook.com>book/151660](https://e.lanbook.com/book/151660)
6. Конева С. И. Функциональное программирование. Ч.1 : учебное пособие Ростов-наДону: СевероКавказский филиал Московского технического университета связи и информатики, 2018, 53 с. 2227-8397, <http://www.iprbookshop.ru/89511.html>
7. Городняя Л. В. Основы функционального программирования Москва: ИнтернетУниверситет Информационных Технологий (ИНТУИТ), 2016, 246 с. 2227-8397, <http://www.iprbookshop.ru/73703.htm>

8.2. Дополнительная литература

1. Кривцова, И. Е. Основы дискретной математики. Часть 1. Учебное пособие [Электронный ресурс] / И. Е. Кривцова, И. С. Лебедев, А. В. Настека. — Электрон. дан. — СПб: ИТМО, 2016. — 92 с. — Режим доступа: http://books.ifmo.ru/book/1869/osnovy_diskretnoy_matematiki_chast_1_uchebnoe_posobie.htm — Загл. с экрана.
2. Душкин, Р. В. Функциональное программирование на языке Haskell / Р. В. Душкин. - 2-е изд. - Москва : ДМК Пресс, 2023. - 609 с. Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". - ISBN 978-5-89818-623-4. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785898186234.html>

8.3. Интернет-ресурсы, необходимые для освоения дисциплины

1. Вики-конспекты. — http://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница
2. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем»: <https://library.asu.edu.ru>
3. Корпоративный проект Ассоциации региональных библиотечных консорциумов (АРБИКОН) «Межрегиональная аналитическая роспись статей» (МАРС): <http://mars.arbicon.ru>
4. Единое окно доступа к образовательным ресурсам <http://window.edu.ru>

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения **лекционных занятий**:

1. Используется аудитория, оборудованная необходимым количеством столов, стульев, доской маркерной и электронной.
2. Аудитория должна иметь следующие нормы освещенности
 - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.
 - СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
3. Электронная доска должна быть подключена к сети Интернет.

Для проведения **лабораторных занятий**:

1. Лабораторные занятия проводятся с группами или подгруппами не более 15 человек.
2. Аудитория должна быть оснащена необходимым количеством столов, стульев, доской маркерной и электронной.
4. Аудитория должна иметь следующие нормы освещенности
 - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.

- СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
5. В аудитории должно быть не менее 15 компьютеров, находящихся в исправном состоянии.
 6. Расположение компьютеров в аудитории должно позволять преподавателю подойти к рабочему месту студента.
 7. Компьютеры должны быть соединены локальной сетью со скоростью не менее 1 Гбит/с и подключены к сети Интернет.
 8. Компьютеры должны обладать минимальными характеристиками:
 - Объем оперативной памяти 16 Гб
 - Накопитель SDD 500 Гб
 - Процессор 12th Gen Intel(R) Core(TM) i3-12100
 - Видеоадаптер Intel(R) UHD Graphics 730

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. Для инвалидов содержание рабочей программы дисциплины (модуля) может определяться также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).