Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО УМС ИИКС

Протокол № 8/1/2025

от 25.08.2025 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ (FUNCTIONAL PROGRAMMING)

Направление подготовки (специальность) [1] 09.03.04 Программная инженерия

Семестр	**Трудоемкость, кред.**	**Общий объем курса, час.**	**Лекции, час.**	**Практич. занятия, час.**	**Лаборат. работы, час.**	**В форме практической подготовки, час.**	**СРС, час.**	**КСР, час.**	**Форма(ы) контроля, экз./зач./КР/КП**
4, 8	2	72	10	20	0		42	0	З
Итого	2	72	10	20	0	0	42	0	

АННОТАЦИЯ

Дисциплина формирует у студентов понимание принципов функционального программирования

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины являются:

– изучение теоретических аспектов функционального программирования на примере абстрактного функционального языка и языка функционального программирования Haskell;

– изучение основ языка Haskell и привитие навыков практической работы с этим языком.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Данная дисциплина является курсом по выбору в цикле дисциплин направления.

Для успешного обучения требуются сложившиеся навыки работы с компьютером и успешное освоение следующие дисциплин:

- информатика;
- дискретная математика (математическая логика);
- дискретная математика (логические исчисления);
- дискретная математика.

Желательно знакомство с основами лямбда-исчисления.

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

Код и наименование компетенции	**Код и наименование индикатора достижения компетенции**

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Задача профессиональной деятельности (ЗПД)	**Объект или область знания**	**Код и наименование профессиональной компетенции; Основание (профессиональный стандарт-ПС, анализ опыта)**	**Код и наименование индикатора достижения профессиональной компетенции**
производственно-технологический			
- освоение и применение средств автоматизированного проектирования, разработки, тестирования и	- программный продукт (создаваемое программное обеспечение) - процессы	ПК-3.1 [1] - Способен к организации разработки программ для небольших и средних по размеру команд разработчиков,	З-ПК-3.1[1] - Знать методы и приемы формализации и алгоритмизации поставленных задач ; У-ПК-3.1[1] - Уметь

сопровождения программного обеспечения; - освоение и применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения; - использование типовых методов для контроля, оценки и обеспечения качества программной продукции; - обеспечение соответствия разрабатываемого программного обеспечения и технической документации российским и международным стандартам, техническим условиям, ведомственным нормативным документам и стандартам предприятия; - участие в процессах разработки программного обеспечения	жизненного цикла программного продукта - методы и инструменты разработки программного продукта	занимающихся созданием программного продукта в условиях неясных или быстро меняющихся требований. *Основание:* Профессиональный стандарт: 06.017	решать при ограниченных временных ресурсах комплексные нестандартные задачи повышенного уровня сложности.; В-ПК-3.1[1] - Владеть различными методологиями разработки программного обеспечения, включая экстремальное программирование.
- освоение и применение средств автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения; - освоение и применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения; -	- программный продукт (создаваемое программное обеспечение) - процессы жизненного цикла программного продукта - методы и инструменты разработки программного продукта	ПК-1 [1] - способен применять основные методы и инструменты разработки программного обеспечения *Основание:* Профессиональный стандарт: 06.001, 06.017	З-ПК-1[1] - Знать основные понятия из области разработки программных систем, применяемые метрики, методы и инструментальные средства.; У-ПК-1[1] - Уметь применять основные методы разработки программного обеспечения; применять основные инструменты разработки программного

использование типовых методов для контроля, оценки и обеспечения качества программной продукции; - обеспечение соответствия разрабатываемого программного обеспечения и технической документации российским и международным стандартам, техническим условиям, ведомственным нормативным документам и стандартам предприятия; - участие в процессах разработки программного обеспечения			обеспечения; В-ПК-1[1] - Владеть основными методами разработки программного обеспечения; основными инструментами разработки программного обеспечения
- освоение и применение средств автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения; - освоение и применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения; - использование типовых методов для контроля, оценки и обеспечения качества программной продукции; - обеспечение соответствия разрабатываемого программного обеспечения и технической	- программный продукт (создаваемое программное обеспечение) - процессы жизненного цикла программного продукта - методы и инструменты разработки программного продукта	ПК-3 [1] - способен применять навыки использования различных технологий разработки программного обеспечения *Основание:* Профессиональный стандарт: 06.001, 06.011, 06.012, 06.022, 06.025	З-ПК-3[1] - Знать различные технологии разработки программного обеспечения; У-ПК-3[1] - Уметь применять различные технологии разработки программного обеспечения; В-ПК-3[1] - Владеть навыками использования различных технологий разработки программного обеспечения

документации российским и международным стандартам, техническим условиям, ведомственным нормативным документам и стандартам предприятия; - участие в процессах разработки программного обеспечения			
научно-исследовательский			
- участие в проведении научных исследований (экспериментов, наблюдений и количественных измерений), связанных с объектами профессиональной деятельности (программными продуктами, проектами, процессами, методами и инструментами программной инженерии), в соответствии с утвержденными заданиями и методиками; - построение моделей объектов профессиональной деятельности с использованием инструментальных средств компьютерного моделирования; - составление описания проводимых исследований, подготовка данных для составления обзоров и отчетов;	- программный проект (проект разработки программного продукта) - программный продукт (создаваемое программное обеспечение) - процессы жизненного цикла программного продукта - методы и инструменты разработки программного продукта	ПК-1.2 [1] - Способен принимать участие в наукоемких программных разработках *Основание:* Профессиональный стандарт: 40.011	З-ПК-1.2[1] - Знать различные методы математического, в первую очередь -- дискретного, моделирования различных объектов и процессов; У-ПК-1.2[1] - Уметь применять методы математического моделирования различных объектов и процессов для разработки и адаптации прикладных моделей; В-ПК-1.2[1] - Владеть методами и средствами создания новых и адаптации существующих прикладных моделей
- участие в проведении научных исследований (экспериментов, наблюдений и количественных	- программный проект (проект разработки программного продукта) -	ПК-11 [1] - способен к формализации в своей предметной области с учетом ограничений используемых методов	З-ПК-11[1] - Знать методы формализации в своей предметной области с учетом ограничений

измерений), связанных с объектами профессиональной деятельности (программными продуктами, проектами, процессами, методами и инструментами программной инженерии), в соответствии с утвержденными заданиями и методиками; - построение моделей объектов профессиональной деятельности с использованием инструментальных средств компьютерного моделирования; - составление описания проводимых исследований, подготовка данных для составления обзоров и отчетов;	программный продукт (создаваемое программное обеспечение) - процессы жизненного цикла программного продукта - методы и инструменты разработки программного продукта	исследования *Основание:* Профессиональный стандарт: 40.011	используемых методов исследования; У-ПК-11[1] - Уметь формализовать в своей предметной области ; В-ПК-11[1] - Владеть методами формализации в своей предметной области с учетом ограничений используемых методов исследования

4. ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ

Направления/цели воспитания	**Задачи воспитания (код)**
Профессиональное воспитание	Создание условий, обеспечивающих, формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (В18)
Профессиональное воспитание	Создание условий, обеспечивающих, формирование научного мировоззрения, культуры поиска нестандартных научно-технических/практических решений, критического отношения к исследованиям лженаучного толка (В19)
Профессиональное воспитание	Создание условий, обеспечивающих, формирование профессионально значимых установок: не производить, не копировать и не использовать программные и технические средства, не приобретённые на законных основаниях; не нарушать признанные нормы авторского права; не нарушать тайны передачи сообщений, не практиковать вскрытие информационных систем и сетей передачи данных; соблюдать конфиденциальность доверенной

	информации (В40)

5. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

№ п.п	Наименование раздела учебной дисциплины	Недели	Лекции/ Практ. (семинары)/ Лабораторные работы, час.	Обязат. текущий контроль (форма*, неделя)	Максимальный балл за раздел**	Аттестация раздела (форма*, неделя)	Индикаторы освоения компетенции
	8 Семестр						
1	Раздел 1	1-5	5/10/0		36	ДЗ-5	З-ПК-1.2, У-ПК-1.2, В-ПК-1.2, З-ПК-3, У-ПК-3, В-ПК-3
2	Раздел 2	6-10	5/10/0		24	ДЗ-10	З-ПК-1.2, У-ПК-1.2, В-ПК-1.2, З-ПК-3, У-ПК-3, В-ПК-3
	Итого за 8 Семестр		10/20/0		60		
	Контрольные мероприятия за 8 Семестр				40	З	З-ПК-3.1, У-ПК-3.1, В-ПК-3.1, З-ПК-1.2, У-ПК-1.2, В-ПК-1.2, З-ПК-1, У-ПК-1, В-ПК-1, З-ПК-3, У-ПК-3, В-ПК-3, З-ПК-11, У-ПК-11, В-ПК-11

* – сокращенное наименование формы контроля

** – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
ДЗ	Домашнее задание
З	Зачет

КАЛЕНДАРНЫЙ ПЛАН

Недели	Темы занятий / Содержание	Лек., час.	Пр./сем., час.	Лаб., час.
	8 Семестр	10	20	0
1-5	**Раздел 1**	5	10	0
1	**Темы 1. Введение в функциональное программирование. Теоретическое основы.** Парадигма программирования. Императивное и декларативное программирование. Функциональное программирование, композиция функций. Формальная система. Элементы лямбда-исчисления (объект в лямбда-исчислении, расстановка скобок, понятие связанной и свободной переменной, применение функции, понятия связанной и свободной переменной, подстановка, конверсия объектов, редукция объектов). Понятие комбинатора, комбинаторная характеристика. Рекурсия в лямбда-исчислении, комбинатор Y. Простое типизированное лямбда-исчисление. Типизация по Чёрчу и по Карри.	Всего аудиторных часов 1 Онлайн 0	2 0	0 0
2	**Тема 2-3. Основы языка Haskell. Рекурсия. Списки.** Свойств языка Haskell (функциональность, лаконичность, ленивость, строгая типизация, модульность). REPL. команды интерпретатора ghci. Константы, переменные, операторы. Синтаксис типов. Ограничения типов. Синтаксис определений, аннотации типов. Основные типы данных: числа, логические значения, списки (синтаксис, генераторы списков, ленивость), кортежи (отличие от списков). Функции: сопоставление с образцом, охранные выражения. Виды рекурсии (функциональная, рекурсивные типы данных). Функциональная рекурсия: рекурсия вместо циклов. Примеры рекурсивных функций. Хвостовая рекурсия. Вычисления с аккумулятором. Списки как рекурсивный тип данных. Функции над списками.	Всего аудиторных часов 1 Онлайн 0	2 0	0 0
3	**Тема 4-5. Каррированные функции. Функции высшего порядка. Производительность рекурсивных алгоритов. Мемоизация.** Способы представления функции от нескольких переменных. Понятие о каррированной функции. Функции высшего порядка. Оператор каррирования. Функции высшего порядка над списками. Лямбда-выражения. Операторы ($) и (.) для упрощения записи выражений.	Всего аудиторных часов 1 Онлайн 0	2 0	0 0

	Производительность функции fib. Принципе разделяй и властвуй, оценка сложности. Динамическое программирование. Мемоизация.			
4	**Тема 6-7. Пользовательские типы данных в языке Haskell. Свёртки.** Синонимы типов. Алгебраические типы данных. Определение пользовательских алгебраических типов данных: конструктор значения типа данных, именованные типы, конструктор типа данных, рекурсивные типы данных. Пример рекурсивных типов данных: бинарные деревья. Определение типа бинарных деревьев, функции добавления элемента, сортировка деревом. Правая свёртка списка. Функции с аккумулятором и левая свёртка списка. Равенство левой и правой свёрток. Префиксные суммы. Свёртка рекурсивного значения рекурсивного типа данных.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
5	**Тема 8-9. Моноиды. Классы типов. Примеры классов типов.** Формальная алгебра. Моноид. Гомоморфизм моноида. Списочный гомоморфизм. Понятие класса типов. Пример класс типа Eq. Определение класса типа. Экземпляр класса типа. Класс типов Monoid. Использование newtype для определения нескольких экземпляров классов типов.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
6-10	**Раздел 2**	5	10	0
6	**Тема 10-11. Категории. Функтор. Аппликативный функтор.** Сведения из теории категорий: объект, морфизм, композиция, тождественный морфизм, функтор. Категория типов Haskell. Эндофунктор в категории типов Haskell. Класс типов для Functor для функторов. Законы функторов. Класс типов Applicative для аппликативных функторов. Законы аппликативного функтора. Использование функтора вместе с аппликативным функтором. ZipList.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
7	**Тема 12-13. Монады. Ввод-вывод.** Класс типов Monad для монад. Связь с фукнтором и аппликативным фукнтором. Стандартные монады: Maybe, List, Reader, Writer, State, IO. do – нотация.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
8	**Тема 14-15. Applicative и MonadPlus. Обработка ошибок. Трансформеры монад. Расширения Haskell.** Классы типов Alternative и MonadPlus. Использование в do-нотации. Тип Either. Обработка ошибок с типом Either. Монада Except. Паргама. Фундепсы. Комбинирование эффектов монад: трансформеры монад. Стандартные трансформеры монад.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
9 - 10	**Тема 16. Доступ к части сложного значения. Линзы.** Понятие линзы к двух функций view и set. Недостаток введённого определения. Функция over. Выражение set через over. Линзы с эффектами. Линзы Ван Лааховена. Реализация функций view, set и over.	Всего аудиторных часов		
		2	4	0
		Онлайн		
		0	0	0

Сокращенные наименования онлайн опций:

Обозначение	Полное наименование
ЭК	Электронный курс
ПМ	Полнотекстовый материал
ПЛ	Полнотекстовые лекции
ВМ	Видео-материалы
АМ	Аудио-материалы
Прз	Презентации
Т	Тесты
ЭСМ	Электронные справочные материалы
ИС	Интерактивный сайт

ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Недели	Темы занятий / Содержание
	8 Семестр
1 - 2	**Элементы лямбда-исчисления.** Задачи на определение связанных и свободных переменны, подстановку, редукцию объектов лямбда-исчисления. Задачи на типизацию объектов лямбда-исчисления: комбинаторы I, K, S, B, C.
3 - 4	**Введение в программирование на языке Haskell.** Примеры работы с интерпретатором ghci: означивание выражений в интерпретатое, создание переменных, вывод типа выражений, загрузка исходного кода в интерпретатор. Программирование функций.
5 - 6	**Программирование на языке Haskell. Списки и рекурсия. Каррирование функций.** Задачи на использование списков, программирование рекурсивных функций: fib, factorial, reverse и т.д. Реализация операторов каррирования и декаррирования для функций 2, 3 и 4-х аргументов. Решения задач о балансировке скобок и списка треугольных чисел.
7 - 8	**Мемоизация. Пользовательские типы данных.** Реализация функции fib при помощи мемоизации. Реализация пользовательского типа данных: пар, комплексный чисел, бинарного дерева.
9 - 10	**Программирование с использованием свёрток. Моноиды.** Реализация функций высшего порядка над списками с использованием свёрток. Реализация аддитивного и мультипликативного моноида для чисел, для логических значений.
11 - 12	**Функтор. Аппликативный функтор.** Реализация функтор и аппликативного функтора для конструкторов типов данных: Maybe, List. Реализация ZipList. Решение задач без использования сравнения с образцом для типов данных, которые поддерживают интерфейс функтора и аппликативного функтора.
13 - 14	**Монады.** Решения задач, требующих чтения, изменения, либо полной поддержки состояний с использованием Monad.
15 - 16	**Ввод-вывод.** Решение задач ввода-вывода с использованием do-нотации и монад.

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основными используемыми образовательными технологиями являются лекции и практические занятия, на которых студенты прослушивают теоретический материал, решают предлагаемые задачи, консультируются с преподавателем относительно семестровых заданий.

Дополнительно на усмотрение преподавателя могут применяться презентационные технологии, предусматривающие использование компьютерной техники как преподавателем, так и студентами. В частности лекции могут сопровождаться демонстрациями слайдов и трансляцией работы преподавателя с интерпретатором функционального языка, а демонстрация готовых семестровых работ может выполняться на ноутбуках студентов.

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Компетенция	**Индикаторы освоения**	**Аттестационное мероприятие (КП 1)**
ПК-1	З-ПК-1	З
	У-ПК-1	З
	В-ПК-1	З
ПК-11	З-ПК-11	З
	У-ПК-11	З
	В-ПК-11	З
ПК-3	З-ПК-3	З, ДЗ-5, ДЗ-10
	У-ПК-3	З, ДЗ-5, ДЗ-10
	В-ПК-3	З, ДЗ-5, ДЗ-10
ПК-3.1	З-ПК-3.1	З
	У-ПК-3.1	З
	В-ПК-3.1	З
ПК-1.2	З-ПК-1.2	З, ДЗ-5, ДЗ-10
	У-ПК-1.2	З, ДЗ-5, ДЗ-10
	В-ПК-1.2	З, ДЗ-5, ДЗ-10

Шкалы оценки образовательных достижений

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	**Оценка по 4-х балльной шкале**	**Отметка о зачете**	**Оценка ECTS**
90-100	5 – *«отлично»*	*«зачтено»*	A

85-89	4 – «*хорошо*»		B
75-84			C
70-74			D
65-69	3 – «*удовлетворительно*»		
60-64			E
ниже 60	2 – «*неудовлетворительно*»	*«не зачтено»*	F

Оценка «отлично» соответствует глубокому и прочному освоению материала программы обучающимся, который последовательно, четко и логически стройно излагает свои ответы, умеет тесно увязывать теорию с практикой, использует в ответах материалы монографической литературы.

Оценка «хорошо» соответствует твердым знаниям материала обучающимся, который грамотно и, по существу, излагает свои ответы, не допуская существенных неточностей.

Оценка «удовлетворительно» соответствует базовому уровню освоения материала обучающимся, при котором освоен основной материал, но не усвоены его детали, в ответах присутствуют неточности, недостаточно правильные формулировки, нарушения логической последовательности.

Отметка «зачтено» соответствует, как минимум, базовому уровню освоения материала программы, при котором обучающийся владеет необходимыми знаниями, умениями и навыками, умеет применять теоретические положения для решения типовых практических задач.

Оценку «неудовлетворительно» / отметку «не зачтено» получает обучающийся, который не знает значительной части материала программы, допускает в ответах существенные ошибки, не выполнил все обязательные задания, предусмотренные программой. Как правило, такие обучающиеся не могут продолжить обучение без дополнительных занятий.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. 004 С32 Функциональное и логическое программирование : учебное пособие для вузов, Волчёнков Н.Г., Сергиевский Г.М., Москва: Академия, 2010

2. ЭИ К88 Функциональное программирование : конспект лекций, Кудрявцев К.Я., Москва: НИЯУ МИФИ, 2020

3. ЭИ Ч-88 Функциональное программирование на С++ : учебное пособие, Чукич И., Москва: ДМК Пресс, 2020

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. 004 Е42 Функциональное и логическое программирование : учебное пособие, Ездаков А.Л., Москва: Бином. Лаборатория знаний, 2011

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Haskell (http://Haskell.org)

https://online.mephi.ru/

http://library.mephi.ru/

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

10. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий,). Каждый раздел проходит аттестацию.

Итоговый балл за каждый раздел составляется из суммы оценок за сданные домашние задания.

Максимальная оценка за домашнее задание составляет 12 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента и зависит от полноты выполнения домашнего задания, а также от сдачи домашнего задания в срок.

В первом разделе курса содержится 3 домашних задания, второй раздел курса включает в себя 2 домашних задания.

Максимальный балл за первый раздел 36 баллов, максимальный балл за второй задел 24 балла.

Баллы за раздел могут быть повышены за посещаемость и активность, но итоговый балл не может превышать максимальный балл за раздел.

Зачёт состоит в решение 2 практических задач на программирование на языке Haskell.

Методические указания по выполнению домашнего задания

За весь курс студент должен выполнить пять домашних работ по соответствующим темам курса. Задания распространяются на семинарских занятиях и в электронном виде по

почте. Актуальное задание и методические указания, если такие требуются, содержатся в тексте самого домашнего задания.

Первые два домашних задания выполняются письменно и требуют предоставление отчёта.

Оформление отчета.

Отчеты оформляются в электронном виде в одном из форматов: Open Document, Microsoft Word, RTF, PDF, PostScript. Отчет должен содержать в себе:

- ФИО студента, номер группы;
- Номер и название домашней работы;
- Решения заданий из домашней работы.

Оставшиеся два домашних задания выполняются в виде файла с кодом на языке Haskell. Шаблон файла с кодом предоставляется вместе с домашним заданием.

Оформления файла с кодом:

- Файла должен именоваться по шаблону: Фамилия_И_О_ГРУППА.hs
- Файл с кодом должен соответствовать шаблону, приложенному к домашнему заданию. Допускается введения дополнительных определений, но не допускается удаление существующих определений или изменение аннотаций типа, а также подключение дополнительных модулей (если обратное не оговорено заданием), кроме тех, что уже подключены в шаблоне файла с кодом.

Оценка работы выставляется исходя из:

- Полноты решения домашнего задания;
- Правильности решения отдельных заданий из домашнего задания.
- Прохождения предоставленным кодом тестов для домашних заданий на языке Haskell.

До 8-й недели студентам даётся билет из 6 задач, общим весом 25 баллов. На 8-й неделе подводится итоговый контроль по результатам сдачи задач.

Пример билета для решения:

На зачёте студенту даётся для решения две практические задачи на языке Haskell.

Примеры возможныз задач:

1. Создание конечных списков из N элементов:

Сконструировать список натуральных чисел. N = 20

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

2. Создание бесконечных списков:

Сконструировать бесконечный список факториалов.

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

3. Построить функцию, вычисляющую N-ый элемент ряда:

F(x, n) = xn

Дать пример вызова построенной функции и ожидаемого результата.

4. Функции работы со списками. Построить функцию:

GetN(L, n) — функция вычленения N-ого элемента из заданного списка

Дать пример вызова построенной функции и ожидаемого результата.

5. Функции с аккумулятором. Написать функцию из задания 3, но с использованием накапливающего параметра.

6. Функции с аккумулятором. Написать функцию из задания 4, но с использованием накапливающего параметра.

7. Сконструировать конечный список из N = 200 натуральных чисел, используя конструктор списка.

Возможные ответы:
[1..200]
или
[1,2..200]

8. Реализуйте функцию, вычисляющую N-ый член ряда Фибоначчи, в виде хвостовой рекурсии.

9. Реализуйте экземляр класса типов Monad для типа данных Either.

10. Реализуйте экземляр класса типов Functor для типа данных Const.

11. Реализуйте экземляр класса типов Functor для типа данных Indentity.

11. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЕЙ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий,). Каждый раздел проходит аттестацию.

Итоговый балл за каждый раздел составляется из суммы оценок за сданные домашние задания.

Максимальная оценка за домашнее задание составляет 12 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента и зависит от полноты выполнения домашнего задания, а также от сдачи домашнего задания в срок.

В первом разделе курса содержится 3 домашних задания, второй раздел курса включает в себя 2 домашних задания.

Максимальный балл за первый раздел 36 баллов, максимальный балл за второй задел 24 балла.

Баллы за раздел могут быть повышены за посещаемость и активность, но итоговый балл не может превышать максимальный балл за раздел.

Зачёт состоит в решение 2 практических задач на программирование на языке Haskell.

Методические указания по выполнению домашнего задания

За весь курс студент должен выполнить пять домашних работ по соответствующим темам курса. Задания распространяются на семинарских занятиях и в электронном виде по почте. Актуальное задание и методические указания, если такие требуются, содержатся в тексте самого домашнего задания.

Первые два домашних задания выполняются письменно и требуют предоставление отчёта.

Оставшиеся два домашних задания выполняются в виде файла с кодом на языке Haskell. Шаблон файла с кодом предоставляется вместе с домашним заданием.

Автор(ы):

Маренков Александр Владимирович

Рословцев Владимир Владимирович