

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
филиал в городе Каире Арабской Республики Египет



УТВЕРЖДАЮ

Ректор

Д.Р. Сафин

2024 г.
МП



Программа дисциплины (модуля)
Функциональное программирование

Направление подготовки: Программная инженерия
Профиль подготовки: Искусственный интеллект и современная разработка программного обеспечения
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: английский
Год начала обучения по образовательной программе: 2025

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-8	<i>Владение навыками использования различных технологий разработки программного обеспечения и систем искусственного интеллекта</i>

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

- грамматики, определяющие абстрактный синтаксис термов бестипового лямбда-исчисления, типов лямбда-исчисления с простыми типами, типов лямбда-исчисления с рекурсивными типами;
- правила вычисления термов бестипового лямбда-исчисления;
- правила типизации термов лямбда-исчисления с простыми типами, лямбда-исчисления с рекурсивными типами;
- алгоритм Хиндли-Милнера вычисления наиболее общего типа;
- кодирование Чёрча чисел, булевых значений, списков;
- простые расширения лямбда-исчисления с простыми типами.

Должен уметь:

- вычислять термы бестипового лямбда-исчисления;
- выводить типы лямбда-исчисления с простыми типами и лямбда-исчисления с рекурсивными типами;
- применять простые расширения лямбда-исчисления с простыми типами;
- программировать интерпретаторы лямбда-исчисления.

Должен владеть:

- теоретическими знаниями о вариантах лямбда-исчисления.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в Блок «Б1.В.ДВ.08.02 Дисциплины, модули» основной профессиональной образовательной программы 09.03.04 - "Программная инженерия (Современная разработка программного обеспечения)" и относится к части, формируемой участниками образовательных отношений.

Осваивается на 4 курсе в 7 семестре(ах).

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины (модуля) составляет 5 зачетных(ые) единиц(ы) на 180 часа(ов).

Контактная работа — 72 часа(ов), в том числе лекции - 0 часа(ов), практические занятия — 0 часа(ов), лабораторные работы - 72 часа(ов), контроль самостоятельной работы — 0 часа(ов).

Самостоятельная работа - 108 часа(ов).

Контроль (зачёт / экзамен) — 0 часа(ов).

Форма промежуточного контроля дисциплины (модуля): зачет с оценкой в 7 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины (модуля)	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Знакомство с функциональным программированием	7	0	0	8	10
2.	Тема 2. Строгая типизация в функциональном языке программирования Haskell	7	0	0	8	10
3.	Тема 3. Рекурсивные типы в Haskell	7	0	0	8	10
4.	Тема 4. Функторы, аппликативные функторы и монады в Haskell	7	0	0	10	18
5.	Тема 5. Бестиповое лямбда-исчисление как модель ФП	7	0	0	12	20
	Тема 6. Простое типизированное лямбда-исчисление	7	0	0	10	16
	Тема 7. Рекурсивные типы в лямбда-исчислении	7	0	0	16	24
	Итого		0	0	72	108

4.2 Содержание дисциплины (модуля)

Тема 1. Знакомство с функциональным программированием

Парадигма функционального программирования как одна из парадигм программирования. Чисто функциональная программа. Отсутствие побочных эффектов. Ленивые и энергичные вычисления, связь с вызовом по имени и по значению. Функции в языке Haskell, их объявление и вызов. Лямбда-функции -- безымянные функции.

Тема 2. Строгая типизация в функциональном языке программирования Haskell

Парадигма функционального программирования как одна из парадигм программирования. Чисто функциональная программа. Отсутствие побочных эффектов. Ленивые и энергичные вычисления, связь с вызовом по имени и по значению. Функции в языке Haskell, их объявление и вызов. Лямбда-функции -- безымянные функции.

Тема 3. Рекурсивные типы в Haskell

Стандартные рекурсивные типы. Тип списка [a] с конструкторами [] и (:). Тип опционального значения Maybe a с конструкторами Nothing и Just. Тип суммы Either a b с конструкторами Left и Right. Использование Either String a для сообщений об ошибках. Изоморфизм типов Maybe a и Either () a. Объявление рекурсивных типов.

Тема 4. Функторы, аппликативные функторы и монады в Haskell

Законы для классов типов. Класс типов Functor, законы класса Functor, примеры типов. Класс типов Applicative, его законы и примеры типов. Класс типов Monad, законы класса Monad и примеры соответствующих типов. Контрпримеры: аппликативный функтор, не являющийся монадой; функтор, не являющийся аппликативным.

Тема 5. Бестиповое лямбда-исчисление как модель ФП

Определение грамматик для языков программирования. Термы бестипового лямбда-исчисления: переменная, лямбда-абстракция, аппликация. Редукция. Правила редукции. Различные стратегии редукции. Нормальная форма. Моделирование булевских констант, натуральных чисел, кортежей в лямбда-исчислении литералами Чёрча.

Тема 6. Простое типизированное лямбда-исчисление

Типы в простом типизированном лямбда-исчислении: базисный тип, тип функции. Определение правил типизации: контекст, правила вывода. Правила типизации простого типизированного лямбда-исчисления. Теорема о строгой нормализуемости: теорема о безопасности и теорема о продвижении. Типизация дополнительных фрагментов (булевы константы и т.п.)

Тема 7. Рекурсивные типы в лямбда-исчислении

Рекурсивные типы. Правила типизации. Примеры рекурсивных типов: "жадные" функции (функции с произвольным количеством аргументов), потоки, объекты. Интерпретация бестипового лямбда-исчисления. Индукция и коиндукция. Эквивалентные и изорекурсивные типы. Регулярные деревья. Ми-типы. Типовые переменные и подстановки. Унификация. Система F.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года №245)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд

оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- критерии оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде – через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;
- в печатном виде – в Научной библиотеке им. Н.И. Лобачевского. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе «Электронный университет». При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину (модуль).

Перечень литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины (модуля). Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. *Haskell Language* - <https://www.haskell.org/>
2. *Hackage: The Haskell Package Repository* - <http://hackage.haskell.org/>
3. *Русскоязычное сообщество Haskell-разработчиков* - <https://ruhaskell.org/>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лабораторные работы	Главная цель лабораторных занятий - осуществить связь теоретических положений с практической действительностью, экспериментальную проверку теоретических положений. Знакомство с оборудованием и выработка навыков работы с ним, уяснение хода выполнения лабораторной работы является обязательным условием

Вид работ	Методические рекомендации
	качественного выполнения работы. Кроме достижения главной цели - подтверждение теоретических положений на лабораторном занятии решаются и другие задачи. При подготовке к лабораторным работам необходимо ознакомиться с методическими указаниями той работы, которая значится в графике учебного процесса и изучить: цель работы; содержание работы; оборудование рабочего места; правила техники безопасности; общие сведения о процессах и режимах установки, стенда, порядок выполнения работы и обработку опытных данных; подготовить отчет о выполненной работе.
самостоятельная работа	В самостоятельную работу входят следующие типы работ: Подготовка к аудиторному занятию (лекция, семинар, лабораторная работа, практическое занятие) и выполнение заданий к нему. Самостоятельное прорабатывание отдельных тем предмета согласно учебно-тематическому плану. Подготовка к практике и выполнение заданий к ней. Подготовка к любым видам контрольных работ. Каждому студенту следует составлять еженедельный и семестровый планы работы, а также план на каждый рабочий день. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подводить итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине это произошло. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана
зачет с оценкой	Для обеспечения систематической и регулярной работы по изучению дисциплины и успешного прохождения промежуточных и итоговых контрольных испытаний студенту рекомендуется придерживаться следующего порядка обучения: 1. Самостоятельно определить объем времени, необходимого для проработки каждой темы. 2. Регулярно изучать каждую тему дисциплины, используя различные формы индивидуальной работы. 3. Согласовывать с преподавателем виды работы по изучению дисциплины.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

- помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КФУ;
- учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной (модулем), за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:
 - продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
 - продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
 - продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Приложение №1
к рабочей программе дисциплины (модуля)
Б1.В.ДВ.10.02. Функциональное программирование

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Казанский (Приволжский) федеральный университет»
Филиал в городе Каире Арабской Республики Египет

Фонд оценочных средств по дисциплине (модулю)
Б1.В.ДВ.08.02 — Функциональное программирование

Направление подготовки: Программная инженерия
Профиль подготовки: Искусственный интеллект и современная разработка программного обеспечения
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: английский
Год начала обучения по образовательной программе: 2025

СОДЕРЖАНИЕ

1. СООТВЕТСТВИЕ КОМПЕТЕНЦИЙ ПЛАНИРУЕМЫМ РЕЗУЛЬТАТАМ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

2. КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

3. РАСПРЕДЕЛЕНИЕ ОЦЕНОК ЗА ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНУЮ АТТЕСТАЦИЮ

4. ОЦЕНОЧНЫЕ СРЕДСТВА, ПОРЯДОК ИХ ПРИМЕНЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ

4.1. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ

4.1.1. Письменное домашнее задание “Язык Haskell”

4.1.1.1. Порядок проведения и процедура оценивания

4.1.1.2. Критерии оценивания

4.1.1.3. Содержание оценочного средства

4.1.2. Контрольная работа “Типы в языке Haskell”

4.1.2.1. Порядок проведения и процедура оценивания

4.1.2.2. Критерии оценивания

4.1.2.3. Содержание оценочного средства

4.1.3. Контрольная работа “Лямбда-исчисление”

4.1.3.1. Порядок проведения и процедура оценивания

4.1.3.2. Критерии оценивания

4.1.3.3. Содержание оценочного средства

4.2. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.2.1. Зачет с оценкой. Устный ответ

4.2.1.1. Порядок проведения и процедура оценивания

4.2.1.2. Критерии оценивания

4.2.1.3. Оценочные средства

4.2.2. Зачет с оценкой. Проверка практических навыков

4.2.2.1. Порядок проведения и процедура оценивания

4.2.2.2. Критерии оценивания

4.2.2.3. Оценочные средства

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций	Оценочные средства текущего контроля и промежуточной аттестации
ПК-8 Владение навыками использования различных технологий разработки программного обеспечения и систем искусственного интеллекта	Знать алгоритм Хиндли-Милнера вычисления наиболее общего типа. Уметь программировать интерпретаторы лямбда-исчисления. Владеть навыками разработки в функциональной парадигме программирования.	Текущий контроль: Письменное домашнее задание «Язык Haskell» Контрольная работа «Типы в языке Haskell» Контрольная работа «Лямбда-исчисление» Промежуточная аттестация: Зачёт с оценкой (Устный ответ, Проверка практических навыков)

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	Средний уровень (хорошо) (71-85 баллов)	Низкий уровень (удовлетворительно) (56-70 баллов)	
ПК-8	<u>Знает</u> алгоритм Хиндли-Милнера вычисления наиболее общего типа	<u>Знает</u> алгоритм Хиндли-Милнера вычисления наиболее общего типа с небольшими ошибками	<u>Знает</u> алгоритм Хиндли-Милнера вычисления наиболее общего типа в общих чертах	<u>Знает</u> Не знает алгоритм Хиндли-Милнера вычисления наиболее общего типа
	<u>Умеет</u> программировать интерпретаторы лямбда-исчисления оптимальным образом	<u>Умеет</u> программировать интерпретаторы лямбда-исчисления с небольшими ошибками	<u>Умеет</u> программировать интерпретаторы лямбда-исчисления в общих чертах	<u>Умеет</u> Не умеет программировать интерпретаторы лямбда-исчисления
	<u>Владеет</u> навыками разработки в функциональной парадигме программирования	<u>Владеет</u> большинством навыков разработки в функциональной парадигме программирования	<u>Владеет</u> некоторыми навыками разработки в функциональной парадигме программирования	<u>Владеет</u> Не владеет навыками разработки в функциональной парадигме программирования

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

7 семестр.

Текущий контроль:

Письменное домашнее задание «Язык Haskell» - 15

Контрольная работа «Типы в языке Haskell» - 15

Контрольная работа «Лямбда-исчисление» - 20

Промежуточная аттестация: зачёт с оценкой.

Зачёт с оценкой проводится по билетам. В билете два задания: первое – устный ответ на вопрос на знание теории курса, второе – практическое задание на проверку приобретенных навыков или формулирование примера по теме.

Устный ответ – 20

Проверка практических навыков – 30

Итого $20 + 30 = 50$ баллов

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: $50+50=100$ баллов.

Соответствие баллов и оценок:

86-100 – отлично

71-85 – хорошо

56-70 – удовлетворительно
0-55 – неудовлетворительно

4. Оценочные средства, порядок их применения и критерии оценивания

4.1. Оценочные средства текущего контроля

4.1.1. Письменное домашнее задание “Язык Haskell”

4.1.1.1. Порядок проведения.

Студенту даётся три задачи для решения на языке Haskell. На решение задач выделяется две недели.

Результатом выполнения задач на написание программы (функции, экземпляра класса и т. п.) является код. Код должен быть написан обучающимся самостоятельно. При написании обучающийся может консультироваться со справочной литературой, но должен мочь объяснить работу полученного кода.

Некоторые задачи также требуют написания доказательств каких-либо характеристик программы, либо доказательств её корректности. Результатом выполнения подобных задач является доказательство, выполненное в любом воспринимаемом преподавателем виде. Обучающийся должен написать доказательство самостоятельно. Доказательство должно быть достаточно строгим в математическом плане, чтобы убедить читающего в правильности доказываемого.

4.1.1.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- пишет код, обеспечивающий разумную для данной задачи временную сложность и сложность по использованию памяти;
- может объяснить структуру и работу собственного решения, доказать соответствующие характеристики сложности решения.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на большинстве допустимых наборов входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- пишет некорректный с синтаксической точки зрения код;
- пишет код, не выдающий правильный результат на всех допустимых наборах входных данных;
- не может объяснить структуру и работу собственное решения.

4.1.1.3. Содержание оценочного средства

Примеры задач письменного домашнего задания «Язык Haskell»:

1. Написать функцию поиска минимума в списке
`getMinimum :: Ord a => [a] -> Maybe a`
2. Написать функцию поиска индекса минимального элемента
`indexMinimum :: Ord a => [a] -> Maybe Int`
3. Написать функцию поиска максимума в списке
`getMaximum :: Ord a => [a] -> Maybe a`
4. Написать функцию поиска индекса максимального элемента
`indexMaximum :: Ord a => [a] -> Maybe Int`
5. Написать функцию поиска суммы элементов списка
`getSum :: Num a => [a] -> a`
6. Написать функцию отбрасывания повторяющихся элементов
`nubOrd :: Ord a => [a] -> [a]`
с асимптотикой $O(N \log N)$
7. Написать функцию сортировки
`sortWith :: Ord b => (a -> b) -> [a] -> [a]`
8. Написать функцию группировки
`groupWith :: Eq b => (a -> b) -> [a] -> [[a]]`
9. Написать функцию группировки
`groupOrd :: Ord a => [a] -> [[a]]`
с асимптотикой $O(N \log N)$
10. Написать функцию `containAllRots :: String -> [String] -> Bool`, которая по заданной строке `strng` и списку строк `arg` выдаёт, содержатся ли всевозможные вращения `strng` в списке `arg`

Например,

```
containAllRots "abc" ["abc", "def", "bca", "cab"] == True
```

```
containAllRots "def" ["abc", "def", "cba", "cab"] == False
```

4.1.2. Контрольная работа “Типы в языке Haskell”

4.1.2.1. Порядок проведения.

Контрольная работа проводится на занятии для проверки теоретической базы студентов. Каждый студент получает случайную задачу. На решение задачи даётся 80 минут, по окончании этого времени собираются тексты решений. При желании студент может сдать своё решение раньше, но после сдачи изменение решения не допускается. Допускается использование компьютера для написания решения.

Результатом выполнения задач на написание программы (функции, экземпляра класса и т. п.) является код. Код должен быть написан обучающимся самостоятельно. При написании обучающийся может консультироваться со справочной литературой, но должен мочь объяснить работу полученного кода.

Некоторые задачи также требуют написания доказательств каких-либо характеристик программы, либо доказательств её корректности. Результатом выполнения подобных задач является доказательство, выполненное в любом воспринимаемом преподавателем виде. Обучающийся должен написать доказательство самостоятельно. Доказательство должно быть достаточно строгим в математическом плане, чтобы убедить читающего в правильности доказываемого.

4.1.2.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- пишет код, обеспечивающий разумную для данной задачи временную сложность и сложность по использованию памяти;
- может объяснить структуру и работу собственного решения, доказать соответствующие характеристики сложности решения.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на большинстве допустимых наборов входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- пишет некорректный с синтаксической точки зрения код;
- пишет код, не выдающий правильный результат на всех допустимых наборах входных данных;
- не может объяснить структуру и работу собственное решения.

4.1.2.3. Содержание оценочного средства

Примеры задач контрольной работы «Типы в языке Haskell»

1. Тип IT задан как $\text{IT } r \ a = (a \rightarrow r \rightarrow r) \rightarrow r \rightarrow r$.

Написать экземпляр класса `Functor` для типа $\text{IT } r$ и показать, что законы `Functor` соблюдаются

2. Тип CT задан как $\text{CT } r \ a = (a \rightarrow r) \rightarrow r$.

Написать экземпляр класса `Functor` для типа $\text{CT } r$ и показать, что законы `Functor` соблюдаются

3. Тип PT задан как $\text{PT } a = \text{String} \rightarrow [(\text{String}, a)]$.

Написать экземпляр класса `Functor` для типа PT и показать, что законы `Functor` соблюдаются

4. Тип IT задан как $\text{IT } r \ a = (a \rightarrow r \rightarrow r) \rightarrow r \rightarrow r$.

Написать экземпляр класса `Applicative` для типа $\text{IT } r$ и показать, что законы `Applicative` соблюдаются

5. Тип CT задан как $\text{CT } r \ a = (a \rightarrow r) \rightarrow r$.

Написать экземпляр класса `Applicative` для типа $\text{CT } r$ и показать, что законы `Applicative` соблюдаются

6. Тип PT задан как $\text{PT } a = \text{String} \rightarrow [(\text{String}, a)]$.

Написать экземпляр класса `Applicative` для типа PT и показать, что законы `Applicative` соблюдаются

7. Тип IT задан как $\text{IT } r \ a = (a \rightarrow r \rightarrow r) \rightarrow r \rightarrow r$.

Написать экземпляр класса `Monad` для типа $\text{IT } r$ и показать, что законы `Monad` соблюдаются

8. Тип CT задан как $\text{CT } r \ a = (a \rightarrow r) \rightarrow r$.

Написать экземпляр класса `Monad` для типа $\text{CT } r$ и показать, что законы `Monad` соблюдаются

9. Тип PT задан как $\text{PT } a = \text{String} \rightarrow [(\text{String}, a)]$.

Написать экземпляр класса `Monad` для типа PT и показать, что законы `Monad` соблюдаются

10. Тип PT задан как $\text{PT } a = \text{String} \rightarrow [(\text{String}, a)]$.

Написать экземпляр класса `Alternative` для типа PT и показать, что законы `Alternative` соблюдаются

11. Для типа выражений

```
data Razor
```

```
  = Lit Int
```

```
  | Add Razor Razor
```

написать интерпретатор следующего типа

```
interpret :: Razor -> Int
```

Функция интерпретации должна проходить, например, следующие тесты:

```
interpret (Add (Lit 1) (Lit 2)) == 3
```

```
interpret (Add (Lit 1) (Add (Lit 2) (Add (Lit 3) (Lit 4)))) == 10
```

12. Для типа выражений

```
data Razor
```

```
  = Lit Int
```

```
  | Add Razor Razor
```

написать функцию, которая выдаёт отформатированное представление выражения

```
pretty :: Razor -> String
```

Функция должна проходить, например, следующие тесты:

```
pretty (Add (Lit 1) (Lit 2)) == "(1+2)"
```

```
pretty (Add (Lit 1) (Add (Lit 2) (Add (Lit 3) (Lit 4)))) == "(1+(2+(3+4)))"
```

4.1.3. Контрольная работа “Лямбда-исчисление”

4.1.3.1. Порядок проведения.

Контрольная работа проводится на занятии для проверки теоретической базы студентов. Каждый студент получает случайную задачу. На решение задачи даётся 80 минут, по окончании этого времени собираются тексты решений. При желании студент может сдать своё решение раньше, но после сдачи изменение решения не допускается.

Результатом выполнения задач на написание программы (функции, экземпляра класса и т. п.) является код. Код должен быть написан обучающимся самостоятельно. При написании обучающийся может консультироваться со справочной литературой, но должен мочь объяснить работу полученного кода.

Некоторые задачи также требуют написания доказательств каких-либо характеристик программы, либо доказательств её корректности. Результатом выполнения подобных задач является доказательство, выполненное в любом воспринимаемом преподавателем виде. Обучающийся должен написать доказательство самостоятельно. Доказательство должно быть достаточно строгим в математическом плане, чтобы убедить читающего в правильности доказываемого.

4.1.3.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- пишет код, обеспечивающий разумную для данной задачи временную сложность и сложность по использованию памяти;
- может объяснить структуру и работу собственного решения, доказать соответствующие характеристики сложности решения.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на всех допустимых наборах входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- пишет корректный с синтаксической точки зрения код;
- пишет код, выдающий правильный результат на большинстве допустимых наборов входных данных;
- может объяснить структуру и работу собственное решения.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- пишет некорректный с синтаксической точки зрения код;
- пишет код, не выдающий правильный результат на всех допустимых наборах входных данных;
- не может объяснить структуру и работу собственное решения.

4.1.3.3. Содержание оценочного средства

Примеры задач контрольной работы «Лямбда-исчисление»

1. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.z) (\lambda s.\lambda z.z)$
2. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.z) (\lambda s.\lambda z.s z)$
3. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.z) (\lambda s.\lambda z.s (s z))$
4. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s z) (\lambda s.\lambda z.z)$
5. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s z) (\lambda s.\lambda z.s z)$
6. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s z) (\lambda s.\lambda z.s (s z))$
7. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s (s z)) (\lambda s.\lambda z.z)$
8. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s (s z)) (\lambda s.\lambda z.s z)$
9. Вычислить $(\lambda m.\lambda n.m (n s) z) (\lambda s.\lambda z.s (s z)) (\lambda s.\lambda z.s (s z))$
10. Написать интерпретатор бестипового лямбда-исчисления.

Например, на заданных входных данных интерпретатор должен выдать следующие выходные данные (с точностью до переименования переменных):

```
(\lambda x.x)(\lambda x.\lambda y.y) -> (\lambda x.\lambda y.y)
```

```
(\lambda x.\lambda y.x)(\lambda x.x)(\lambda x.\lambda y.y) -> (\lambda x.x)
```

$\lambda s.\lambda z.(\lambda m.\lambda n.\lambda s.\lambda z.m (n s z) z)(\lambda s.\lambda z.s (s (s z)))(\lambda s.\lambda z.s (s z)) s z \rightarrow (\lambda s.\lambda z.s (s (s (s z))))$
 $(\lambda x.x x)(\lambda x.x x) \rightarrow (\lambda x.x x)(\lambda x.x x)$

11. Написать функцию вывода типов в системе F.

4.2. Оценочные средства промежуточной аттестации

4.2.1. Зачет с оценкой. Устный ответ

4.2.1.1. Порядок проведения.

Студент дает устный ответ на первый вопрос билета. Время на подготовку билета – 40 минут. Студент может делать записи при подготовке к ответу и пользоваться ими при ответе, однако чтение ответа по листку бумаги не принимается. Не допускается использование каких-либо источников информации, кроме билета. Преподаватель выслушивает устный ответ студента, задает дополнительные и уточняющие вопросы.

4.2.1.2. Критерии оценивания.

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- Свободно владеет основными понятиями.
- Дает полный ответ на вопрос.
- Демонстрирует высокую подготовленность и эрудицию.
- Приводит свои или не озвученные на занятиях примеры.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- Владеет основными понятиями.
- Дает ответ на вопрос.
- Допускает небольшие погрешности и неточности.
- Приводит общеизвестные или озвученные на занятиях примеры.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- В целом, владеет основными понятиями.
- Дает неполный ответ на вопрос.
- Допускает погрешности и неточности.
- Не все примеры уместны.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- Слабо владеет основными понятиями.
- Не дает ответ на вопрос.
- Не приводит примеры.

4.2.1.3. Оценочные средства.

Примеры типовых заданий для устного ответа на зачёте с оценкой:

1. Термы и значения бестипового лямбда-исчисления
2. Правила вычисления бестипового лямбда-исчисления
3. Кодирование булевых констант в бестиповом лямбда-исчислении
4. Кодирование пар в бестиповом лямбда-исчислении
5. Кодирование натуральных чисел в бестиповом лямбда-исчислении
6. Термы лямбда-исчисления с простыми типами
7. Значения в лямбда-исчисления с простыми типами
8. Типы лямбда-исчисления с простыми типами
9. Правила вычисления лямбда-исчисления с простыми типами
10. Правила типизации лямбда-исчисления с простыми типами
11. Добавление булевых констант в лямбда-исчисление с простыми типами
12. Добавление пар в лямбда-исчисление с простыми типами
13. Добавление натуральных чисел в лямбда-исчисление с простыми типами
14. Теорема о продвижении в лямбда-исчислении с простыми типами
15. Теорема о безопасности в лямбда-исчислении с простыми типами
16. Термы лямбда-исчисления с рекурсивными типами
17. Значения в лямбда-исчисления с рекурсивными типами
18. Типы лямбда-исчисления с рекурсивными типами
19. Правила вычисления лямбда-исчисления с рекурсивными типами
20. Правила типизации лямбда-исчисления с рекурсивными типами

4.2.2. Зачет с оценкой. Проверка практических навыков

4.2.2.1. Порядок проведения.

Часть проверки практических навыков включает в себя одно практическое задание. Студенту необходимо описать требуемое понятие и привести примеры использования.

Перечисленные типовые задания позволяют провести проверку навыков по соответствующим темам учебного курса.

Студент выполняет практическое задание письменно. Задание является второй частью билета. Объем ответа в среднем может составлять до 3 страниц. В ответе учитывается форматирование и структура подачи информации. Использование блок-схем приветствуется.

4.2.2.2. Критерии оценивания.

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- В работе соблюдены все требования.
- Документ структурирован, поддается изучению.
- В выборе алгоритмов и компонентов программы ошибок нет.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- В работе соблюдена большая часть требований.
- Документ структурирован, поддается изучению. Имеется несколько ошибок.
- В выборе алгоритмов и компонентов программы допущено несколько ошибок.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- В работе соблюдена некоторая часть требований.
- Документ плохо структурирован. Имеется много ошибок.
- В выборе алгоритмов и компонентов программы допущено множество ошибок.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- Требования не соблюдены.
- В документе отсутствует структура. Имеется много ошибок.
- Выбор алгоритмов и компонентов программы неуместен или не осуществлен.

4.2.2.3. Оценочные средства.

Примеры типовых заданий для проверки практических навыков на зачёте с оценкой:

1. Чистые функции, отсутствие эффектов в Haskell
2. Типы выражений в Haskell
3. Типы функций в Haskell
4. Каррирование в Haskell
5. Описание алгебраических типов данных (АТД) в Haskell
6. Описание АТД, параметризованных типами, в Haskell
7. Описание рекурсивных АТД в Haskell
8. Классы типов, класс типов Eq,Ord,Show,Read
9. Класс типов Functor, его законы и примеры
10. Класс типов Applicative, его законы и примеры
11. Класс типов Monad, его законы и примеры
12. Тип Cont как представитель классов Functor, Applicative, Monad
13. Примеры типов, являющихся экземплярами класса типов Applicative, но не Monad
14. Примеры типов, являющихся экземплярами класса типов Functor, но не Applicative
15. Примеры типов, допускающих несколько вариантов реализации Applicative
16. Примеры типов, допускающих несколько вариантов реализации Monad
17. Стандартный тип Maybe a и функции для работы с ним
18. Стандартный тип Either a b и функции для работы с ним
19. Стандартный тип (a,b) и функции для работы с ним
20. Стандартные функции для работы со списками: map, filter, foldl, foldr

Пример билета:

1. Термы и значения бестипового лямбда-исчисления
2. Стандартные функции для работы со списками: map, filter, foldl, foldr

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: Программная инженерия

Профиль подготовки: Искусственный интеллект и современная разработка программного обеспечения

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: английский

Год начала обучения по образовательной программе: 2025

Основная литература:

1. Масленникова, О.Е. Основы искусственного интеллекта : учебное пособие / О.Е. Масленникова, И.В. Гаврилова. — 3-е изд., стер. — Москва : ФЛИНТА, 2019. — 283 с. - ISBN 978-5-9765-1602-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1034902> (дата обращения: 12.12.2022). - Режим доступа: по подписке.

2. Мински Я., Программирование на языке OCaml: учебное пособие / Мински Я., Мадхавapedди А., Хикки Дж. – Москва: ДМК Пресс, 2014. - 536 с. - ISBN 978-5-97060-102-0 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785970601020.html> (дата обращения: 12.12.2022). - Режим доступа : по подписке.

3. Томпсон С., Программирование в Erlang: учебное пособие / Томпсон С., Чезарини Ф. ; Пер. с англ. Холoméва А. О. – Москва: ДМК Пресс, 2012. - 488 с. (Серия "Функциональное программирование") - ISBN 978-5-94074-617-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785940746171.html> (дата обращения: 12.12.2022). - Режим доступа : по подписке.

Дополнительная литература:

1. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1735805> (дата обращения: 12.12.2022). - Режим доступа : по подписке.

2. Душкин, Р. В. Функциональное программирование на языке Haskell / Душкин Р. В. - Москва : ДМК Пресс, 2008. - ISBN 5-94074-335-8. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN5940743358.html> (дата обращения: 12.12.2022). - Режим доступа : по подписке.

3. Эмерик Ч., Программирование на Clojure: монография / Эмерик Ч., Карпер Б., Гранд К. – Москва: ДМК Пресс, 2015. - 816 с. - ISBN 978-5-97060-299-7 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785970602997.html> (дата обращения: 12.12.2022). - Режим доступа : по подписке.

Перечень информационных технологий, включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: Программная инженерия

Профиль подготовки: Искусственный интеллект и современная разработка программного обеспечения

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: английский

Год начала обучения по образовательной программе: 2025

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

1. Операционная система Microsoft office professional plus 2010, или Microsoft Windows 7 Профессиональная, или Windows XP (Volume License)
2. Пакет офисного программного обеспечения Microsoft Office 365, или Microsoft office professional plus 2010
3. Adobe Reader XI или Adobe Acrobat Reader DC
4. Браузер Mozilla Firefox
5. Браузер Google Chrome
6. Kaspersky Endpoint Security для Windows
7. Программная система для обнаружения текстовых заимствований в учебных и научных работах. АО «Антиплагиат»
8. Электронная библиотечная система «ZNANIUM.COM»
9. Электронная библиотечная система Издательства «Лань»
10. Электронная библиотечная система «Консультант студента»