**Программа учебной дисциплины “Функциональное программирование”**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор | Степан Львович Кузнецов, к.ф.-м.н. |
| Число кредитов | 4 |
| Контактная работа (час.) | 46 |
| Самостоятельная работа (час.) | 106 |
| Курс | 4-й курс бакалавриата |
| Формат изучения дисциплины | без использования онлайн-курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины “Функциональное программирование” являются освоение теоретической базы функционального программирования (начала теории типов, лямбда-исчисление, начала теории категорий); овладение техникой программирования на языке Haskell.

Ожидается, что студент, успешно освоивший дисциплину, знает теоретические основы функционального программирования (лямбда-исчисление, начала теории категорий и проч.); умеет решать задачи на программирование на языке Haskell; умеет разрабатывать новые библиотеки на языке Haskell; владеет общей информацией об императивном и функциональном стилях программирования, их сильных и слабых сторонах, умеет выбирать правильный стиль в зависимости от задачи; умеет создавать код на функциональном языке, корректность которого формально доказана (верифицирована). Настоящая дисциплина относится к профессиональному циклу, дисциплинам базовой части профиля. Для специализации «Распределенные системы» настоящая дисциплина является базовой. Изучение данной дисциплины базируется на знании курса “Сложность вычислений и логика в теоретической информатике”.

# Содержание УЧЕБНОЙ ДИСЦИПЛИНЫ

*Тема 1. Типизованное лямбда-исчисление*

Это теоретический вводный раздел курса. В нём на лекции излагаются основные понятия лямбда-исчисления, понятия, теории типов и их применения в программировании. На семинаре обсуждаются задачи на типизацию и преобразования лямбда-термов.

*Тема 2. Нормализуемость и конфлюэнтность*

На лекции даются формулировки основных свойств лямбда-исчисления (точнее, отношений бета- и ита-редукций): свойства сильной и слабой нормализуемости и свойство Чёрча - Россера (конфлюэнтности). Доказательства этих свойств выходят за рамки курса. На семинаре обсуждаются примеры нормализации лямбда-термов, а также нарушения нормализуемости (как слабой, так и сильной, при сохранении слабой) в нетипизируемом случае.

*Тема 3. Императивное и функциональное программирование. Элементы функционального программирования в императивных языках*

На лекции даётся сравнение двух парадигм программирования - императивной и функциональной. Практическое занятие и домашняя работа посвящены реализации некоторых мотивов функциональной парадигмы во в целом императивном языке Python.

*Тема 4. Язык и среда разработки Haskell: установка и настройка*

Практическое занятие: демонстрация работы с интерпретатором Haskell (GHC или иным).

*Тема 5. Основы программирования на языке Haskell*

Практическое занятие: простейшие программы на Haskell.

*Тема 6. Типы данных в Haskell*

Даётся обзор стандартных типов данных в Haskell. Обсуждаются возможности индуктивного построения новых типов (на примере натуральных чисел по Пеано). Обсуждается алгоритм Хиндли-Дамаса-Милнера для выведения типов (type inference).

*Тема 7. "Ленивые" и "ретивые" вычисления*

На лекции обсуждается различие между "ленивыми" вычислениями (когда значение подтерма вычисляется только по необходимости) и "ретивыми" вычислениями (когда рекурсивно вычисляется всё). На практическом занятии даются примеры, когда "ленивые" вычисления справляются с ситуациями, где "ретивые" входят в бесконечный цикл.

*Тема 8. Монады и их использование для ввода-вывода*

На лекции рассказывается введение в теорию категорий и даётся понятие монады. На практическом занятии обсуждается применение монад для реализации функций с состояниями и побочных эффектов.

*Тема 9\*. Соответствие Карри - Говарда. Верификация программ и математических утверждений. Система Coq*

Необязательная часть курса. Рассказывается о соответствии Карри - Говарда между типами и высказываниями и термам и доказательствами высказываний. В практической части демонстируются возможности формального доказательства математических теорем, в том числе о правильности работы программ (верификация).

# ОЦЕНИВАНИЕ

Текущий контроль освоения дисциплины проводится посредством программистских домашних заданий (4-6 заданий в течение модуля) и контрольной работы (60 мин) в середине курса. Итоговый контроль состоит в письменном экзамене (90 мин) после окончания курса. Домашние задания могут раздаваться и приниматься по электронной почте, посредством системы Github или каким-либо иным удобным для преподавателя и студентов способом. В ходе текущего контроля (домашние задания) студент должен продемонстировать владение техниками программирования, обсуждёнными к настоящему моменту. На самостоятельной работе и итоговом экзамене - показать владение теоретическими основами функционального программирования, посредством решения задач. Оценки по всем формам текущего и итогового контроля выставляются по 10-балльной шкале. Итоговая оценка формируется следующим образом: 0,7\*(накоп.) + 0,3\*(экзамен), где (экзамен) - оценка за итоговый экзамен, а (накоп.) - накопленная оценка за семестр, определяемая следующим образом: (накоп.) = 0,72\*(прог.) + 0,28\*(сам.), где (прог.) - суммарная оценка за программистские домашние задания, (сам.) - оценка за самостоятельную работу. Округление при выставлении окончательной оценки осуществляется в большую сторону (в пользу студента). Блокирующих элементов нет. Студент, имеющий накопленную оценку 9 или 10, имеет право засчитать её в качестве итоговой и не участвовать в экзамене. Для студентов с накопленной оценкой 8 эта возможность остаётся на усмотрение преподавателя.

# ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

Примеры оценочных заданий приведены на web-странице <http://www.mi-ras.ru/~sk/lehre/fp_hse2018/>

1. **РЕСУРСЫ**
   1. **Основная литература**

А. Холомьёв. Учебник по Haskell. 2012. Электронный ресурс: <https://anton-k.github.io/ru-haskell-book/files/ru-haskell-book.pdf>

Д. Шевченко. О Haskell по-человечески. 2016. Электронный ресурс: <https://www.ohaskell.guide/pdf/ohaskell.pdf>

H. Barendregt. Lambda calculus with types. (<http://ttic.uchicago.edu/~dreyer/course/papers/barendregt.pdf>)

M. Lipovača. Learn you a Haskell for the great good, 2011.(<http://learnyouahaskell.com/>)

G. Hutton. Programming in Haskell (2nd ed.). Cambridge Univ. Press, 2016 (<https://people.southwestern.edu/~potter/HaskellCode/hutton.pdf>)

* 1. **Дополнительная литература**

Damas, Luis; Milner, Robin (1982), "Principal type-schemes for functional programs", POPL '82: Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on principles of programming languages (<http://groups.csail.mit.edu/pag/6.883/readings/p207-damas.pdf>)

В. Н. Крупский, С. Л. Кузнецов. Компьютерный практикум по математической логике (Coq.) МГУ, 2013. <http://www.mi.ras.ru/~sk/lehre/coq/coq_pract.pdf>

Kees Doets, Jan van Eijck. The Haskell Road to Logic, Maths and Programming. College Publications, 2004. (<https://fldit-www.cs.uni-dortmund.de/~peter/PS07/HR.pdf>)

* 1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Glasgow Haskell Compiler | *Свободный доступ (на условиях лицензии типа BSD): https://www.haskell.org/ghc/* |
| 2. | Haskell Stack | *Свободный доступ (на условиях лицензии типа BSD): http://www.haskellstack.org/* |
| 3. | Coq/CoqIDE | *Свободный доступ (на условиях лицензии GNU LGPL): http://coq.inria.fr/* |

* 1. **Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  | ***Профессиональные базы данных, информационно-справочные системы*** | |
| 1. | Hackage: архив пакетов Haskell | URL: https://hackage.haskell.org/ |
| 2. | Документация Haskell | URL: https://www.haskell.org/documentation |
| 3. | GitHub: веб-сервис для хостинга IT-проектов и совместной разработки | URL: https://github.com/ |
|  | ***Интернет-ресурсы (электронные образовательные ресурсы)*** | |
| 1. | Вики-система факультета компьютерных наук НИУ ВШЭ | URL: http://wiki.cs.hse.ru/ |
| 2. | Bartosz Milewski’s Programming Cafe | URL: https://bartoszmilewski.com/ |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории оснащены мультимедийным проекторами для демонстрации примеров программного кода, а также стандартной доской с мелом или маркерами.