МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета информационных технологий

/Д. Г. Демидов/

«28» 04 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Функциональное и логическое программирование»

Направление подготовки

09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль подготовки)

«Программное обеспечение информационных систем»

Квалификация (степень) выпускника

бакалавр

Форма обучения

заочная

Москва 2022

Программа дисциплины «Функциональное и логическое программирование» составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению **09.03.01 Информатика и вычислительная техника** и профилю подготовки **«Программное обеспечение информационных систем».**

Программу составил 
_____/О.В.Дедехина/

Программа дисциплины утверждена на заседании кафедры «Прикладная информатика»
«28» августа 2022 г. протокол № 1
Заведующий кафедрой
доцент, к.э.н.  /С. В. Суворов/

Программа согласована с руководителем образовательной программы по направлению подготовки **09.03.01 Информатика и вычислительная техника** по профилю подготовки **«Программное обеспечение информационных систем».**


_____/С. В. Суворов/
«____» августа 2022 г.

Программа утверждена на заседании учебно-методической комиссии факультета Информационных технологий

Председатель комиссии  / Д. Г. Демидов/

«____»_____2022 г. Протокол:

1. Цели освоения дисциплины

Основные цели дисциплины «Функциональное программирование»:

- формирование представления о функциональном программировании;
- выработка практических навыков применения функционального подхода при разработке программ;
- освоение функционального языка программирования Haskell.

Основные задачи дисциплины «Функциональное программирование»:

- изложение теоретических основ функционального программирования и их применения в современном программировании;
- изучение рекурсивных алгоритмов и их реализация;
- знакомство с функциональным подходом к программированию, приобретение навыков программирования на языке функционального программирования Haskell;
- подготовка теоретической базы для дальнейшего углубленного изучения отдельных вопросов в специализированных разделах математической логики и функционального программирования.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Функциональное программирование» (Б.1.2.11) относится к числу профессиональных учебных дисциплин вариативной части базового цикла основной образовательной программы бакалавриата.

К началу изучения курса студенты должны овладеть теоретическими знаниями, полученными при изучении следующих курсов: «Компьютерная математика» (Б.1.2.5), «Информатика» (Б.1.1.10), «Программирование» (Б.1.1.15), «Методы хранения и обработки информации» (Б.1.2.9), «ЭВМ и периферийные устройства» (Б.1.2.2), «Сети и телекоммуникации» (Б.1.2.3). Необходимо, чтобы студенты имели практические навыки программирования и знакомство с основными функциями операционных систем и вычислительных сетей.

Знания, навыки и умения, приобретенные в результате прохождения курса, также могут быть востребованы студентами при подготовке выпускной квалификационной работы бакалавра.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции, и ими должны быть достигнуты следующие результаты обучения (как этап формирования соответствующих компетенций):

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать	Перечень планируемых результатов обучения по дисциплине
ПК-1	Обладать способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина».	**знать:** основные принципы функционального программирования; основы лямбда-исчисления; основы теории комбинаторов; методы реализации функциональных языков; теорию рекурсивных функций и лямбда-исчисление А.Черча; различия между энергичными и

		ленивыми языками функционального программирования; соответствие между функциональными и императивными программами; **уметь:** применять полученные знания для решения конкретных прикладных задач; ориентироваться в современных языках функционального программирования, их возможностях; обосновывать выбор языка функционального программирования для решения конкретных задач; обосновывать выбор представления и обработки данных для решения поставленной задачи; **владеть:** методами и приемами программирования с использованием функциональных языков; навыками разработки рекурсивных функций

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет **5** зачетные единицы, т.е. **180** академических часа, из них **164** часов – самостоятельная работа студентов. Все они осваиваются обучающимися в седьмом семестре (на четвертом курсе).

Виды учебных занятий по дисциплине: лекции – 1 час в неделю (всего 4 часа), лабораторные занятия – **1** час в неделю (всего **12** часов). Форма контроля – зачет.

Содержание разделов дисциплины

Принципы функционального программирования

Основные характеристики функциональных языков программирования. Энергичное и ленивое вычисление. Язык функционального программирования Haskell. Язык функционального программирования ML. Метод структурной индукции.

Лямбда-исчисление

Синтаксис и семантика лямбда-исчисления. Связывание переменных и подстановка. Правила преобразования. Редукция. Теорема Черча-Россера. Ромбическое свойство. Порядок редукций. Рекурсивные выражения. Взаимная рекурсия. Определение комбинаторов фиксированной точки. Заголовочная и слабая заголовочная нормальные формы. Чистое лямбда-исчисление.

Методы интерпретации функциональных языков программирования

Методы интерпретации функциональных языков. Промежуточный код для интепретатора функционального языка. Энергичный Eval/Apply интепретатор. Ленивый Eval/Apply интерпретатор. Энергичная SECD-машина. Ленивая SECD-машина.

Комбинаторы и редукция графов

Представление лямбда-выражений в виде графов. Правила редукции графов. Проблема свободных переменных. Основы комбинаторной логики и редукции. Методы компиляции с использованием комбинаторов. G-машина.

Структура и содержание дисциплины представлены в приложении 1 к рабочей программе.

5. Образовательные технологии

Методика преподавания дисциплины «Функциональное программирование» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

- чтение лекций;
- проведение лабораторных работ;
- проведение практических занятий;
- проведение регулярных устных опросов.

Темы лабораторных и практических работ:

- «Функциональность. Введение в функциональное программирование на языке функционального программирования Haskell»;
- «Лямбда-исчисление. Формы рекурсии. Функции высшего порядка в языке Haskell. Композиция функций»;
- «Методы интерпретации функциональных языков. Соответствие между функциональными и императивными программами»;
- «Комбинаторы и редукция графов».

Удельный вес занятий, проводимых в интерактивных формах, определен главной целью образовательной программы, особенностью контингента обучающихся и содержанием дисциплины «Функциональное программирование» и в целом по дисциплине составляет 25% аудиторных занятий. Занятия лекционного типа составляют 50% от объема аудиторных занятий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- проверка домашних заданий;
- проверка готовности студентов к проведению лабораторных работ;
- проверка выполненных лабораторных работ;
- проведение экзамена.

Примерные вопросы к экзамену приведены в приложении 2.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-1	Обладать способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина».

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины (модуля), описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ПК-1 – обладать способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина»

Показатель	Критерии оценивания			
	2	3	4	5
знать: основные принципы функционального программирования; основы лямбда-исчисления; основы теории комбинаторов; методы реализации функциональных языков; теорию рекурсивных функций и лямбда-исчисление А.Черча; различия между энергичными и ленивыми языками функционального программирования, соответствие между функциональными и императивными программами	Обучающийся не знает основные принципы функционального программирования.	Обучающийся знает основные принципы функционального программирования.	Обучающийся знает основы лямбда-исчисления, основы теории комбинаторов, методы реализации функциональных языков; теорию рекурсивных функций и лямбда-исчисление А.Черча.	Обучающийся знает различия между энергичными и ленивыми языками функционального программирования, соответствие между функциональными и императивными программами.
уметь: применять полученные знания для решения конкретных прикладных задач; ориентироваться в	Обучающийся не умеет применять полученные знания для решения конкретных прикладных задач.	Обучающийся умеет применять полученные знания для решения конкретных прикладных задач.	Обучающийся умеет ориентироваться в современных языках функционального программирования,	Обучающийся умеет обосновывать выбор представления и обработки данных для решения

современных языках функционального программирования, их возможностях; обосновывать выбор языка функционального программирования для решения конкретных задач; обосновывать выбор представления и обработки данных для решения поставленной задачи			их возможностях; обосновывать выбор языка функционального программирования для решения конкретных задач.	поставленной задачи.
владеть: методами и приемами программирования с использованием функциональных языков; навыками разработки рекурсивных функций	Обучающийся не владеет методами программирования с использованием функциональных языков.	Обучающийся владеет методами программирования с использованием функциональных языков.	Обучающийся владеет приемами программирования с использованием функциональных языков.	Обучающийся владеет навыками разработки рекурсивных функций.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: экзамен

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине, при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Функциональное программирование» (выполнили практические и лабораторные работы).

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент в основном демонстрирует соответствие знаний, умений, навыков

	приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены некоторые ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Удовлетворительно	Студент демонстрирует удовлетворительное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются умеренные ошибки, проявляется неполное наличие знаний, умений, навыков по ряду показателей, студент испытывает затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Фонд оценочных средств представлен в приложении 2 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины

Основная литература:

1. Зыков С. В. Введение в теорию программирования. Функциональный подход. [Электронный ресурс] – Электрон. дан. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 153 с. – Режим доступа: http://www.knigafund.ru/books/197652/read#page1

Дополнительная литература:

1. Ефимова Е. А. Основы программирования на языке Visual Prolog. [Электронный ресурс] – Электрон. дан. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 266 с. – Режим доступа: http://www.knigafund.ru/books/177042/read#page1

Программное обеспечение и интернет-ресурсы:

1. Свободное программное обеспечение, входящее в базовую поставку ОС Linux: Браузер Mozilla Firefox, компилятор и интерпретатор языка Haskell, текстовый редактор Atom.
2. Офисные приложения LibreOffice для Linux (свободное ПО)
3. Офисные приложения Microsoft Office 2013(или ниже) - Microsoft Open License. Лицензия № 61984042
4. Microsoft office 2013 prof (для обучения). Госконтракт № 18-09/14 от 22.09.2014 Акт № Тг09950
5. Интерпретатор языка Ruby (свободное ПО)
6. Компилляторы gcc, gcc+ (свободное ПО)
7. Система контроля версий Git (свободное ПО)

8. Материально-техническое обеспечение дисциплины

Компьютерные классы с ОС Linux в аудиториях: ав1201, ав1202, оснащенная: Компьютеры, столы, стулья, аудиторная доска, проектор. Рабочее место преподавателя: компьютер, стол, стул.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины «Функциональное программирование» осуществляется в строгом соответствии с целевой установкой рабочей программы по дисциплине. При самостоятельной работе студентам рекомендуется в первую очередь прорабатывать лекционные материалы, дополняя их сведениями из тематических литературы и информационных ресурсов. Теоретические знания закрепляются посредством выполнения лабораторных работ и решения практических задач в рамках аудиторных занятий, к которым требуется своевременная самостоятельная подготовка. Для углубления получаемых знаний и выработки исследовательских навыков студенту предлагается выполнить ряд домашних заданий и изучить отдельные темы. Важным элементом освоения студентом дисциплины является его стремление к систематизации знаний, получаемых по всем видам данной дисциплины, а также выстраивание логических связей между данной дисциплиной и дисциплинами изученными ранее. При возникновении у студента вопросов локального характера по материалам дисциплины преподавателем дистанционно, с помощью современных средств телекоммуникации, оказывается консультационная помощь.

10. Методические указания для преподавателя

Проведение занятий по дисциплине «Функциональное программирование» осуществляется в строгом соответствии с целевой установкой и в тесной взаимосвязи с учебным планом. Основой теоретической подготовки студентов являются лекции. При рассмотрении учебных материалов рекомендуется делать акцент на практические примеры, демонстрировать их реальную работу с помощью проектора.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты лекций, готовятся к экзамену, а также самостоятельно изучают отдельные темы учебной программы.

Важным обстоятельством является привлечение внимания студентов к обсуждаемой проблеме, стимулирование интереса к ней и организация активного обсуждения, как структуры проблемы, так и составляющих ее наиболее актуальных тем. Для повышения эффективности проведения занятия требуется предварительная подготовка всех его участников. В этой связи рекомендуется заблаговременно (не менее, чем за неделю) оповестить студентов о теме занятия, дать перечень литературы по теме.

При проведении практического занятия преподаватель выполняет, в основном, функции ведущего – направляет студентов в правильное русло решения задач, рассматривает оптимальность предложенных решений, корректирует возможные ошибки.

Активная работа студента на практическом занятии учитывается при определении итоговой оценки его знаний по дисциплине на экзамене.

Самостоятельная работа по дисциплине «Функциональное программирование» предполагает: выполнение студентами домашних заданий. Домашние задания являются, как правило, продолжением практических занятий и содействуют овладению практическими навыками по основным разделам дисциплины. Самостоятельная работа студентов предполагает изучение теоретического и практического материала по актуальным вопросам дисциплины. Рекомендуется самостоятельное изучение учебной и научной литературы, использование справочной литературы и др.

При выдаче заданий на самостоятельную работу используется дифференцированный подход к студентам. Перед выполнением студентами

самостоятельной внеаудиторной работы преподаватель проводит инструктаж по выполнению задания, который включает: цель здания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. В процессе инструктажа преподаватель предупреждает студентов о возможных типичных ошибках, встречающихся при выполнении задания. Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Текущий контроль осуществляется на практических занятиях, промежуточный контроль осуществляется на экзамене в письменной или устной форме.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность умений;
- оформление материала в соответствии с требованиями.

Приложение 1

Структура и содержание дисциплины «Функциональное программирование» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (бакалавр)

п/п	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб.	СРС	КСР	К.Р.	К.П.	РГР	Реферат	К/р	Э	З
1	**Шестой семестр**														+
1.1	Принципы функционального программирования	7	1	1		3	41	+							+
1.2	Лямбда-исчисление	7	6	1		3	41	+							+
1.3	Методы интерпретации функциональных языков программирования	7	9	1		3	41	+							+
1.4	Комбинаторы и редукция графов	7	11	1		3	41	+							+
	Форма аттестации		**18-21**												+
	Всего часов по дисциплине в шестом семестре			4		12	164								

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Направление подготовки:
09.03.01 «Информатика и вычислительная техника»

Профиль подготовки
«Информатика и вычислительная техника»

Форма обучения: очная
Вид профессиональной деятельности: проектно-конструкторская

Кафедра: Прикладная информатика

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

«Функциональное программирование»

Состав:

1. Паспорт фонда оценочных средств
2. Перечень оценочных средств
3. Оценочные средства

Москва 2022 г.

Паспорт фонда оценочных средств по дисциплине «Функциональное программирование» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (бакалавр)

«Функциональное программирование»					
ФГОС ВО 09.03.01 «Информатика и вычислительная техника» (уровень бакалавриата)					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **профессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства**	**Степени уровней освоения компетенций**
Индекс	**Формулировка**				
ПК-1	способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина»	**знать:** основные принципы функционального программирования; основы лямбда-исчисления; основы теории комбинаторов; методы реализации функциональных языков; теорию рекурсивных функций и лямбда-исчисление А.Черча; различия между энергичными и ленивыми языками функционального программирования; соответствие между функциональными и императивными программами; **уметь:** применять полученные знания для решения конкретных прикладных задач; ориентироваться в современных языках функционального программирования, их возможностях; обосновывать выбор языка функционального	лекции, лабораторные работы, практические занятия	экзамен (Экз)	**пороговый уровень:** владеет методами и приемами программирования с использованием функциональных языков; **базовый уровень:** умеет применять полученные знания для решения конкретных прикладных задач; **повышенный уровень:** умеет ориентироваться в современных языках функционального программирования, их возможностях; обосновывать выбор языка функционального программирования для решения конкретных задач

		программирования для решения конкретных задач; обосновывать выбор представления и обработки данных для решения поставленной задачи; **владеть:** методами и приемами программирования с использованием функциональных языков; навыками разработки рекурсивных функций			

Перечень оценочных средств по дисциплине «Функциональное программирование» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (бакалавр)

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Экзамен (Экз)	Средство промежуточной аттестации студента, проводится в письменно-устной форме.	Перечень вопросов по темам (разделам) дисциплины.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы

Примерный перечень вопросов к экзамену по дисциплине «Функциональное программирование» (ПК-1):

1. Понятие функциональности (прозрачность по ссылкам).
2. Методы функционального программирования: структурная индукция.
3. Методы функционального программирования: энергичное и ленивое вычисления.
4. Стили функционального программирования. Язык функционального программирования Standard ML.
5. Стили функционального программирования. Язык функционального программирования Haskell.
6. Чистое лямбда-исчисление.
7. Синтаксис и семантика лямбда-исчисления.
8. Связывание переменных и подстановка в лямбда-исчислении.
9. Правила преобразования и редукция в лямбда-исчислении.
10. Теорема Черча-Россера.
11. Ромбическое свойство.
12. Порядок редукций в лямбда-исчислении.
13. Представление рекурсивных выражений в лямбда-исчислении.
14. Комбинатор фиксированной точки.
15. Заголовочная и слабая заголовочная нормальные формы.
16. G-машина.
17. Промежуточный код для интепретатора функционального языка. Энергичный Eval/Apply интепретатор.
18. Промежуточный код для интепретатора функционального языка. Ленивый Eval/Apply интерпретатор.
19. Энергичная SECD-машина.
20. Ленивая SECD-машина.
21. Представление лямбда-выражений в виде графов.
22. Правила редукции графов.
23. Основы комбинаторной логики.
24. Комбинаторная редукция.

25. Методы компиляции с использованием комбинаторов.
26. Представление рекурсии с использованием комбинаторов.
27. Базовые типы языка Haskell.
28. Определение новых классов данных в Haskell. Конструкторы.
29. Задание синонимов для типов данных в Haskell. Примеры.
30. Свертки. Примеры использования.
31. Абстракции (генераторы) списков в Haskell. Примеры.
32. Рекурсивные типы данных. Бинарное дерево поиска. Сортировка списков с помощью дерева поиска.
33. Ленивость языка Haskell и бесконечные структуры данных. Бесконечные списки. Функции на бесконечных списках.
34. Функции высшего порядка. Отображение (map). Примеры использования.
35. Частичная параметризация. Карринг.
36. Строгие конструкторы при создании типов данных.
37. Модули. Списки экспорта и импорта.
38. Композиция функций в Haskell.
39. Операторы: ассоциативность и приоритет. Задание операторов.
40. Перегрузка функций. Полиморфизм функций.
41. Рекурсия и техника накопления параметров.
42. Определение деревьев и функций их обработки.
43. Каррирование и декаррирование функций.
44. Полиморфные типы в языке Haskell. Примеры.
45. Стандартные классы в Haskell: Eq, Ord, Num, Show и другие. Определение экземпляров (instance).
46. Монада IO и функции ввода-вывода в Haskell.
47. Монада Maybe в Haskell и её использование.
48. Определение функций. Сопоставление с образцом.
49. Лямбда-абстракции в Haskell (\p1...pn -> e). Сечения функций (inc = (+) 1).
50. Списки, конструкторы списков, кортежи в Haskell.