Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ИС*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Функциональное программирование

Направление подготовки *09.03.02 Информационные системы и технологии*

Профиль подготовки *Информационные системы и технологии*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
7	**72 / 2**	**16**	**16**		**1,6**	**0,25**	**33,85**	**38,15**	**Зач.**
Итого	**72 / 2**	**16**	**16**		**1,6**	**0,25**	**33,85**	**38,15**	

Муром, 2026 г.

1. Цель освоения дисциплины

Целью освоения дисциплины «Функциональное программирование» является обучение студентов концепциям и методам функционального программирования на примере языка Haskell.

Задачи дисциплины:

В результате освоения курса «Функциональное программирование» студенты должны иметь представление:

- О стилях программирования, преимуществах и недостатках функционального подхода;
- О принципах рекурсивного перебора списков;
- О функциях высших порядков;
- Об основных типах данных языка и операторах Haskell;
- О способах разработки собственных структур данных.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Функциональное программирование» обеспечивает понимание основ функционального программирование и написания программ на языке Haskell. Курс базируется на знаниях, полученных студентами в процессе изучения дисциплин «Информатика», «Математика», «Технологии программирования».

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-5 Способность выполнять работы по созданию (модификации) и сопровождению информационных систем	ПК-5.1 Применяет современные средства создания, модификации и сопровождения информационных систем	Применяет современные средства создания, модификации и сопровождения информационных систем (ПК-5.1)	Устный опрос,Вопросы при защите работы, контрольная работа,Вопросы при защите практической работы,Контрольная работа

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетных единицы, 72 часа.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Концепции функционального программирования	7	2								устный опрос
2	Основы Haskell	7	4	4						18	отчет по ПР и промежуточная КР
3	Функции в Haskell	7	4	4							отчет по ПР
4	Функции высшего порядка	7	2	4						4	отчет по ПР
5	Числовые функции	7	2							2	промежуточная КР
6	Структуры данных	7	2	4						14,15	отчет по ПР
Всего за семестр		72	16	16				1,6	0,25	38,15	Зач.
Итого		72	16	16				1,6	0,25	38,15	

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 7

Раздел 1. Концепции функционального программирования

Лекция 1.

Тема 1. Концепции функционального программирования (2 часа).

Раздел 2. Основы Haskell

Лекция 2.

Тема 2. Основы Haskell: Компилятор GHCi, Базовые типы Haskell (2 часа).

Лекция 3.

Тема 2. Основы Haskell: Символы. Списки, Охранные выражения (2 часа).

Раздел 3. Функции в Haskell

Лекция 4.

Тема 3. Функции в Haskell: Полиморфизм и перегрузка (2 часа).

Лекция 5.

Тема 3. Функции в Haskell: Операторы, Карринг (2 часа).

Раздел 4. Функции высшего порядка

Лекция 6.

Тема 4. Функции высшего порядка (2 часа).

Раздел 5. Числовые функции

Лекция 7.

Тема 5. Числовые функции (2 часа).

Раздел 6. Структуры данных

Лекция 8.

Тема 6. Структуры данных (2 часа).

4.1.2.2. Перечень практических занятий

Семестр 7

Раздел 2. Основы Haskell

Практическое занятие 1

Основы работы с Haskell (2 часа).

Практическое занятие 2

Основы работы с Haskell (2 часа).

Раздел 3. Функции в Haskell

Практическое занятие 3

Рекурсия и работа со списками (2 часа).

Практическое занятие 4

Рекурсия и работа со списками (2 часа).

Раздел 4. Функции высшего порядка

Практическое занятие 5

Функции высшего порядка (2 часа).

Практическое занятие 6

Функции высшего порядка (2 часа).

Раздел 6. Структуры данных

Практическое занятие 7

Разработка собственных типов данных (2 часа).

Практическое занятие 8

Разработка собственных типов данных (2 часа).

4.1.2.3. Перечень лабораторных работ

Не планируется.

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Транслятор Hugs.
2. Консольный ввод-вывод.
3. Графический интерфейс в Haskell.
4. Классы в Haskell.
5. Лямбда-исчисление.
6. Алгебраические типы данных.
7. Рекурсивные типы.
8. Инкапсуляция данных.
9. Модули.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

4.2 Форма обучения: заочная

Уровень базового образования: среднее профессиональное.
Срок обучения 3г 6м.

Семестр	Трудоемкость, час./ зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль,час.	Всего (контактная работа), час.	СРС, час.	Переаттестация	Форма промежуточного контроля (экз., зач., зач. с оц.)
2	**72 / 2**	**2**	**2**		**1**	**0,5**	**5,5**	**26,75**	**36**	**Зач.(3,75)**
Итого	**72 / 2**	**2**	**2**		**1**	**0,5**	**5,5**	**26,75**	**36**	**3,75**

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Концепции функционального программирования	2	2							0	Тест
2	Основы Haskell	2		2						0	Отчет по ПР
3	Функции в Haskell	2								11	Тест
4	Функции высшего порядка	2								10	Тест
5	Числовые функции	2								3	Тест
6	Структуры данных	2								2,75	Тест
Всего за семестр		36	2	2		+		1	0,5	26,75	Зач.(3,75)
Итого		36	2	2				1	0,5	26,75	3,75
Итого с переаттестацией		72									

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 2

Раздел 1. Концепции функционального программирования

Лекция 1.

Концепции функционального программирования. Основы Haskell. Функции в Haskell. Функции высшего порядка (2 часа).

4.2.2.2. Перечень практических занятий

Семестр 2

Раздел 2. Основы Haskell

Практическое занятие 1.

Основы работы с Haskell (2 часа).

4.2.2.3. Перечень лабораторных работ

Не планируется.

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Транслятор Hugs.
2. Числовые функции.
3. Структуры данных.
4. Консольный ввод-вывод.
5. Графический интерфейс в Haskell.
6. Классы в Haskell.
7. Модули.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

1. Обзор языков функционального программирования.
2. Типы данных в Haskell.
3. Обзор языка LISP.
4. Обзор трансляторов Haskell.
5. История развития функциональных языков.
6. Обзор языка Miranda.
7. Основные концепции функционального программирования.
8. Монады Haskell.
9. λ-исчисление.
10. Обзор языка Erlang.
11. Функции высших порядков.
12. Типичные задачи, решаемые функциональным программированием.
13. Механизм вывода типов.
14. Обзор языка ML.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

В процессе изучения дисциплины «Функциональное программирование» применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении практических работ применяется имитационный или

симуляционный подход, когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Рогозин О.В. Функциональное и рекурсивно-логическое программирование. - М.: Евразийский открытый институт, 2012. - 139 c. - http://www.iprbookshop.ru/11119.html

2. Дорохова, Т. Ю. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Ю. Дорохова, И. Е. Ильина. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 139 c. — ISBN 978-5-4488-1531-7, 978-5-4497-1718-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. - https://www.iprbookshop.ru/122426.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Учебно-методическое пособие по дисциплине Логическое и функциональное программирование [Электронный ресурс]/ — Электрон. текстовые данные.— М.: Московский технический университет связи и информатики, 2012.— 23 c. - http://www.iprbookshop.ru/61490.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

- электронная библиотечная системы "IPRBooks" (http://www.iprbookshop.ru/);

Программное обеспечение:
Erlang (Apache License 2.0)
Haskell (Haskell EULA)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория ГИС и САПР

Сервер; 12 персональных компьютеров; проектор Sanyo PDG-DSU20; экран настенный Drapper Apex Star

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Обучающиеся выполняют индивидуальную задачу в соответствии с заданием на практическую работу. Полученные результаты сводятся в отчет и защищаются по традиционной методике на следующем практическом занятии.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – зачет. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.02 Информационные системы и технологии* и профилю подготовки *Информационные системы и технологии*
Рабочую программу составил *к.т.н, доцент Комкова С.В.*_____

Программа рассмотрена и одобрена на заседании кафедры *ИС*

протокол № 21 от 05.05.2026 года.
Заведующий кафедрой *ИС* _____*Андрианов Д.Е.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Функциональное программирование

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

ПК-5

Вопросы для устного опроса

1. Основы функционального программирования.
2. Типы данных в Haskell.
3. Особенности языка LISP.
4. Трансляторы Haskell.
5. История развития функциональных языков.
6. Основные концепции функционального программирования.
7. Монады Haskell.
8. λ-исчисление.
9. Функции высших порядков.
10. Типичные задачи, решаемые функциональным программированием.
11. Механизм вывода типов.

Задания на промежуточную контрольную работу №1:

Привести примеры выражений, соответствующих типу

1. [(Integer, [Char], String)]
2. (([Double], [Bool]), Char)
3. ((Integer, [Double]), [String])
4. [([String], Bool, Double)]

Реализовать на Haskell две функции из перечисленных ниже

1. Функция, принимающая на вход координаты трёх точек и возвращающая максимальное расстояние между ними.
2. Функция, принимающая на вход три целых числа и возвращающая сумму квадратов тех из них, которые являются положительными.
3. Функция, принимающая на вход координаты трех точек и возвращающая True, если они лежат в одном квадранте координатной сетки и False в противном случае.
4. Функция, принимающая на вход координаты центра круга, его диаметр и координаты точки и возвращающая пару (b,d), где b – True, если точка лежит внутри круга, False – если она лежит вне его; d – расстояние от точки до границы окружности.

Задания на промежуточную контрольную работу №2:

Реализовать на Haskell две функции из перечисленных ниже

1. Функция, возвращающая максимальный чётный элемент списка. Если чётных элементов в списке нет – возвращать -1.
2. Функция, возвращающая минимальный нечётный элемент списка. Если нечётных элементов в списке нет – возвращать -1.
3. Функция, возвращающая из списка точек те из них, которые лежат на координатных осях.
4. Функция, формирующая список значений из указанного диапазона с указанным шагом. Границы диапазона могут быть заданы в порядке убывания.
5. Функция, принимающая список вещественных чисел и возвращающая пару из максимального и минимального значений в списке
6. Функция, принимающая на вход координаты трёх точек и возвращающая максимальное расстояние между ними.
7. Функция, принимающая на вход три целых числа и возвращающая сумму квадратов тех из них, которые являются положительными.

8. Функция, принимающая на вход координаты трех точек и возвращающая True, если они лежат в одном квадранте координатной сетки и False в противном случае.

9. Функция, принимающая на вход координаты центра круга, его диаметр и координаты точки и возвращающая пару (b,d), где b – True, если точка лежит внутри круга, False – если она лежит вне его; d – расстояние от точки до границы окружности.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Устный опрос (2 вопроса), контрольная работа	20
Рейтинг-контроль 2	контрольная работа, защита практических работ	20
Рейтинг-контроль 3	Защита практических работ	20
Посещение занятий студентом	Отметка в журнале посещений	10
Дополнительные баллы (бонусы)		10
Выполнение семестрового плана самостоятельной работы	Защита практических и контрольных работ	20

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

ПК-5

Для проведения зачетной работы используются задания в тестовой форме, приведённые далее (в разделе 3).

Методические материалы, характеризующие процедуры оценивания

При проверке знаний, приобретенных в рамках выполнения практических работ, используются контрольные вопросы, приведенные в методических указаниях к работам. Защита практических работ является средством промежуточной аттестации.

Всего необходимо защитить восемь практических работ и написать две контрольные работы как минимум на оценку "удовлетворительно".

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***

66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

Функциональное программирование

Вопросы открытого типа

1. Как в Haskell записать символ кавычек (")

A) """

B) '\"'

C) ""

D) "\""

ANSWER: B

2. Как в Haskell записать список значений

A) list(1,2,3)

B) {1,2,3}

C) [1,2,3]

D) (1,2,3)

ANSWER: C

3. Как объявить функцию, принимающую вещественное число и возвращающую пару целых чисел

A) func :: Double -> (Int, Int)

B) func :: Double -> Int -> Int

C) func :: Double

D) > (Int)

E) func :: Double -> [Int]

ANSWER: D

4. Как объявить функцию, принимающую строку и целое число и возвращающую целое число

A) func :: String -> Int -> Int
B) func :: [Char] :: Int -> Int
C) func :: String -> Int
D) func :: [Char] -> Int
E) > Int
ANSWER: E

5. Какой тип будет иметь выражение [('a', 'b'), (1, 2)]

A) выражение составлено с ошибкой
B) [(Char, Char), (Int, Int)]
C) [(Char), (Int)]
D) [(Char, Int)]
ANSWER: A

6. Какой результат вернёт следующее выражение: head [[1, 2, 3]]

A) 1
B) []
C) [1,2,3]
D) [2,3]
E) [1]
ANSWER: C

7. Какой результат вернёт следующее выражение: tail [[1, 2, 3]]

A) 1
B) []
C) 3
D) [2,3]
E) [3]
ANSWER: B

8. Что называется лямбда-функцией?

A) Функция, принимающая в качестве аргумента или возвращающая в качестве результата другую функцию
B) Функция, не имеющая имени и определяемая внутри другой функции
C) Группа функций, вызываемых последовательно одна за другой
D) Функция, которая может принимать аргументов меньше, чем у неё объявлено формальных параметров
ANSWER: B

9. Какой тип будет иметь выражение (1, 2, "abc")

A) выражение составлено с ошибкой
B) ([Int], [Char])
C) (Int, String)
D) (Int, Int, String)
E) (Int, Int, Char)
ANSWER: D

10. Какой тип будет иметь выражение [('a', 1), ('b', 2)]

A) выражение составлено с ошибкой
B) [(Char, Int), (Char, Int)]
C) [(Char), (Int)]
D) [(Char, Int)]
ANSWER: D

11. Отметьте название языка программирования, в котором впервые поддержаны идеи функционального программирования

A) Lisp
B) Scheme
C) Haskell

ANSWER: A

12. Какой метод решения задач особенно характерен для функционального программирования?

A) преобразование состояний памяти

B) обработка сигналов

C) символьная обработка информации

ANSWER: C

13. Какой из перечисленных механизмов реализации способствовал применению функционального программирования неспециалистами?

A) полнота средств управления вычислениями

B) простота и гибкость отладки

C) отсутствие "синтаксического сахара" в представлении программ

ANSWER: B

14. Стандартизированный чистый функциональный язык программирования общего назначения:

A) Haskell

B) Scala

C) Ruby

ANSWER: A

15. Одна из основных характеристик языка Haskell:

A) ленивые вычисления

B) резкие вычисления

C) мгновенные вычисления

ANSWER: A

Вопросы закрытого типа

1. Как в Haskell возвести число 3 в степень 5

(3^5)

2. Какой результат вернёт следующее выражение: length [[1, 2, 3], [4, 5]]

(2)

3. Квадратный корень в Haskell вычисляется с помощью функции

(sqrt)

4. Оператор целочисленного деления в Haskell

(div)

5. Сколько значений может возвращать функция в концепции функционального программирования

(1)

6. Функция проверки числа на чётность в Haskell

(even)

7. Чему будет равно значение функции f (x + 2) = x - 3 при вызове f (10-5)

(0)

8. Какая функция в Haskell возвращает первый элемент списка

(head)

9. Какая функция в Haskell возвращает список без первого элемента

(tail)

10. Какая функция в Haskell возвращает длину списка

(lenght)

11. Какая функция в Haskell возвращает последний элемент списка

(last)

12. Как в Haskell записать символьный литерал 3

(‘3’)

13. Чему в Haskell будет равно значение выражения 7/10

(0.7)

14. Функция, принимающая в качестве аргумента или возвращающая в качестве результата другую функцию называется функцией ….

(высшего порядка)

15. В каком языке программирования впервые реализованы идеи функционального программирования? Ответ на английском языке.

(Lisp)

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/question/edit.php?cmid=6106&category=35346%2C18571&qbshowtext=0&qbshowtext=1&recurse=0&showhidden=0

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.