

Министерство науки и высшего образования Российской Федерации
 Федеральное государственное бюджетное образовательное учреждение
 высшего образования
 «Владимирский государственный университет
 имени Александра Григорьевича и Николая Григорьевича Столетовых»
 (ВлГУ)

УТВЕРЖДАЮ

Проректор
 по образовательной деятельности



А.А. Панфилов

« 03 » 09

2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

(наименование дисциплины)

Направление подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Профиль/программа подготовки

Уровень высшего образования

бакалавриат

Форма обучения

очная ускоренное

| Семестр | Трудоемкость зач. ед./ час. | Лекции, час. | Практич. занятия, час. | Лаборат. работы, час. | СРС, час. | Форма промежуточного контроля (экз./зачет) |
|---------|--------------------------------|-----------------|------------------------------|-----------------------------|--------------|---|
| 5 | 6/216 | 36 | - | 36 | 117 | Экзамен (27), КР |
| Итого | 6/216 | 36 | - | 36 | 117 | Экзамен (27), КР |

Владимир 2018

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Основной целью курса является формирование и закрепление системного подхода при разработке программ с применением методов и языков функционального программирования.

Ядро курса составляют теоретические основы, а также инструментарии создания программ с использованием декларативных языков функционального программирования.

Для успешного изучения курса студенту необходимо знать основы математической логики.

В курсе закрепляются такие общепредметные умения, как выбор языка программирования для решения поставленной задачи, выбор способа представления исходных данных и выбор метода решения поставленной задачи.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Функциональное программирование» относится к дисциплинам по выбору вариативной части блока Б1 ОПОП подготовки бакалавров по направлению «Математическое обеспечение и администрирование информационных систем».

Дисциплина изучается в пятом семестре и требует освоения следующих курсов:

- Алгоритмы и алгоритмические языки;
- Объектно-ориентированное программирование;
- Структуры и алгоритмы компьютерной обработки данных;
- Базы данных;
- Иностранный язык.

В рамках перечисленных дисциплин студенты получают следующие знания и умения, необходимые для освоения курса «Функциональное программирование»:

- Умение применять методики алгоритмизации задач, выбирать наиболее эффективные алгоритмы;
- Знание и навыки использования объектно-ориентированного подхода при разработке программных продуктов;
- Знание основных современных средств разработки программных продуктов;
- Умение применять системный подход и основные методы проектирования программных продуктов;
- Знание теоретических основ и технологий, навыки использования современных систем управления базами данных;
- Знание моделей компьютерных сетей, принципов их функционирования;
- Умение получать информацию из источников на иностранном языке.

Дисциплина формирует знания и навыки, необходимые в практической деятельности квалифицированного специалиста. В рамках учебного процесса может быть использована при подготовке выпускной квалификационной работы.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины студент должен частично овладеть следующими компетенциями и продемонстрировать следующие результаты образования:

| Коды компетенции | Результаты освоения ООП | Перечень планируемых результатов обучения |
|------------------|---|--|
| ОПК-3 | готовностью анализировать проблемы и направления развития технологий программирования | <p>Знать: проблемы и направления развития технологий программирования.</p> <p>Уметь: анализировать проблемы и направления развития технологий программирования.</p> <p>Владеть: знаниями о направлениях развития технологий программирования.</p> |
| ОПК-4 | способностью применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения | <p>Знать: среды разработки программного обеспечения с применением функционального подхода.</p> <p>Уметь: использовать в профессиональной деятельности средства разработки программного обеспечения с применением функциональных языков.</p> <p>Владеть: навыками использования стандартных и разработки новых алгоритмических и программных решений с применением основ функционального программирования, современными средствами разработки, поддерживающими функциональные языки.</p> |
| ОПК-5 | владением информацией о направлениях развития компьютеров с традиционной (нетрадиционной) архитектурой; о тенденциях развития функций и архитектур проблемно-ориентированных программных систем и комплексов | <p>Знать: историю развития функциональных языков программирования, тенденции развития области функциональных языков.</p> <p>Уметь: работать с информацией из различных источников, включая сетевые ресурсы сети Интернет, для решения профессиональных задач в области функционального программирования</p> <p>Владеть: информацией о направлениях развития компьютеров.</p> |
| ОПК-7 | способностью использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений. | <p>Знать: современные языки функционального программирования и среды разработки.</p> <p>Уметь: осуществлять разработку программного продукта используя современные инструментальные средства, создания программного обеспечения.</p> <p>Владеть: современными инструментальными средствами, создания программного обеспечения.</p> |

| Коды компетенции | Результаты освоения ООП | Перечень планируемых результатов обучения |
|------------------|---|---|
| ПК-2 | готовностью к использованию основных моделей информационных технологий и способов их применения для решения задач в предметных областях | <p>Знать: основные модели информационных технологий и особенности разработки приложений на функциональных языках.</p> <p>Уметь: осуществлять разработку программного продукта используя современные инструментальные средства, создания программного обеспечения с учетом особенностей разработки на функциональных языках.</p> <p>Владеть: современными инструментальными средствами создания программного обеспечения, поддерживающие разработку на функциональных языках.</p> |
| ПК-3 | готовностью к разработке моделирующих алгоритмов и реализации их на базе языков и пакетов прикладных программ моделирования | <p>Знать: особенности функционального программирования и среды разработки, применительно к моделирующим алгоритмам.</p> <p>Уметь: разрабатывать типовые моделирующие алгоритмы с применением функциональной парадигмы программирования.</p> <p>Владеть: современными инструментальными средствами, создания программного обеспечения на функциональных языках.</p> |

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

| № п/п | Раздел (тема) дисциплины | Семестр | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | | | Объем учебной работы, с применением интерактивных методов (в часах / %) | Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам) |
|-------|--|---------|-----------------|--|----------------------|---------------------|--------------------|-----|---------|---|---|
| | | | | Лекции | Практические занятия | Лабораторные работы | Контрольные работы | СРС | КП / КР | | |
| 1 | Раздел 1. Введение в функциональное программирование | 5 | 1-2 | 4 | - | 2 | - | 9 | + | - | |
| 2 | Раздел 2. Лямбда-исчисление Чёрча. | 5 | 3-5 | 8 | - | 4 | - | 20 | + | 6 / 50% | |
| 3 | Раздел 3. Принципы функционального программирования. | 5 | 6-10 | 8 | - | 10 | - | 24 | + | 6 / 33% | Рейтинг-контроль №1 |
| 4 | Раздел 4. Языки функционального программирования. | 5 | 11-14 | 8 | - | 16 | - | 36 | + | 12 / 50% | Рейтинг-контроль №2 |

| № п/п | Раздел (тема) дисциплины | Семестр | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | | | Объем учебной работы, с применением интерактивных методов (в часах / %) | Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам) |
|--------------------|---|---------|-----------------|--|----------------------|---------------------|--------------------|-----|---------|---|---|
| | | | | Лекции | Практические занятия | Лабораторные работы | Контрольные работы | СРС | КП / КР | | |
| 5 | Раздел 5. Интерпретация языков функционального программирования. | 5 | 15-16 | 6 | - | 4 | - | 21 | + | 6 / 60% | |
| 6 | Раздел 6. Применения функционального программирования. Перспективы развития | 5 | 17-18 | 2 | | - | | 7 | | - | Рейтинг-контроль №3 |
| Итого за 5 семестр | | 5 | 18 | 36 | - | 36 | - | 117 | КР | 30 / 41% | Экзамен (27), КР |

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Темы лекций

1. Введение в дисциплину. Понятие о функциональном программировании, его сравнение с императивным программированием и место в решении задач искусственного интеллекта.
2. Лямбда-исчисление Чёрча. Тезис Чёрча-Тьюринга. Алфавит, утверждения, аксиомы и правила вывода в лямбда-исчислении. Конвертируемость и редукция лямбда-выражений. Нормальная форма.
3. Лямбда-исчисление Чёрча. Рекурсии в лямбда-исчислении. Чистое лямбда-исчисление.
4. *Рейтинг-контроль №1 на лекции.* Принципы функционального программирования: Функции высших порядков. Рекурсивные функции. Статическое и динамическое связывание.
5. Принципы функционального программирования: Отложенные вычисления. Понятие о строгих функциональных языках.
6. Принципы функционального программирования: Карринг.
7. *Рейтинг-контроль №2 на лекции.*
8. Языки функционального программирования: Основы языка Haskell. Символы, константы, логические значения. Списки и способы их записи. Кортежи.
9. Базовые функции языка. Передача параметров. Область действия параметров.
10. Определение функции в программе. Простая рекурсия. Работа со списками.
11. Haskell как язык ООП: встроенные классы, примеры структурированных типов данных, инкапсуляция данных в модулях.
12. Ввод и вывод информации: монады, интерактивный ввод/ вывод информации, работа с файлами.
13. Интерпретация языков функционального программирования. SECD-машина.
14. Построение интерпретаторов для SECD-машины. Реализация энергичного и отложенного интерпретаторов.
15. Метод редукции графов.
16. Применения функционального программирования в решении задач искусственного интеллекта.
17. Современный этап развития функционального программирования.
18. *Рейтинг-контроль №3 на лекции.*

Лабораторный практикум (контроль освоения разделов 1 - 5)

Состоит из семи лабораторных работ.

Список тем лабораторных занятий:

- 1) Язык функционального программирования Haskell. Основные типы данных и конструкции языка.
- 2) Язык Haskell. Рекурсии в функциональном программировании. Сопоставление с образцом.
- 3) Язык Haskell. Функции высших порядков.
- 4) Язык Haskell. Конструкции `let` и `where`. Охраняющие условия. Полиморфизм. Пользовательские типы данных.
- 5) Язык Haskell. Рекурсивные типы данных. Операторы.
- 6) Язык Haskell. Модули. Операции ввода-вывода.
- 7) Язык Haskell. Графический интерфейс (GUI).

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В рамках лекционного курса:

- Технология проблемного обучения (case study). При рассмотрении вопросов практического применения рассмотренного теоретического материала, используется диалог со студентами на предмет возможных способов решения поставленной задачи. Особенно активно данная технология применяется в рамках третьего и четвертого раздела курса.

В рамках лабораторного практикума:

- Технология уровневой дифференциации. Прежде всего, при постановке заданий на лабораторные работы по третьему разделу курса, а также при проведении контрольного мероприятия по этому разделу и при проведении экзамена.

В рамках курсовой и самостоятельной работы:

- Метод проектов. В группах по 2-3 человека студенты на практике осваивают большую часть этапов жизненного цикла разработки программного обеспечения, начиная с постановки задачи и заканчивая внедрением.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Примерный перечень вопросов и заданий рейтинг-контроля:

Рейтинг-контроль №1 (контроль освоения разделов 1-2)

1. Показать, что $m + n = \lambda f x. m f (n f x)$, если $n = \lambda f x. f^n x$ и $m, n \in N$.
2. Свободные переменные.
3. Форма Бэкуса-Наура определения лямбда - терма.
4. Понятие нормальной формы λ - термов.
5. Пары и кортежи: определение и функции извлечения компонентов пары.
6. Постфиксная форма именования выражений (where)?
7. Синтаксическая эквивалентность термов с точностью до α - конверсии. Как обозначается? Привести примеры.
8. Можно ли определить рекурсивные функции без именования? Каким образом?

Рейтинг-контроль №2 (контроль освоения раздела 3)

1. Приведите причины, по которым следует придерживаться принципов строгой типизации.
2. Определите функцию `cube`, возводящую в куб числа типа `Float`.
3. Наибольший приоритет имеет функция или оператор ?
4. Отличия списка от кортежа.
5. Что такое охраняемые выражения? Привести пример использования.
6. Дать рекурсивное определение функции 2^n .
7. Укажите тип следующих величин: 3, `even` и `even 3`.
8. Если в объявлении функции `f` содержится три стрелочки, то `f x` будет содержать.
9. Назначение функции `filter`. Привести пример применения.
10. Дать определение композиции функций. Как обозначена операция композиции в Haskell.
11. Конструкторы создания списков.
12. Вычислите `map (map (^2)) [[1,2],[3,4,5]]`.
13. Задайте бесконечный список натуральных чисел.

Рейтинг-контроль №3 (контроль освоения раздела 4-6)

1. Объявить класс с методами сложения и взятия обратного элемента. Определить экземпляр класса для типа `Int`.
2. Объявление экземпляра класса `instance` означает
3. Назначение класса `Eq`. Какие методы определены в классе.
4. Ключевое слово для определения наследования.
5. Определить тип-синоним месяца года.
6. Написать программу, запрашивающую с клавиатуры ввод названия входного и выходного файла и осуществляющую копирование информации из одного в другой файл.
7. Мутирующая рекурсия в языке `ISWIM`.
8. Перечислите состояния `SECD`-машины.
9. Стек в `SECD`-машине. Назначение.
10. Пояснить понятие «Монада».
11. Что означают операции `(>>=)` и `(>>)`, определенные в классе `Monad`.
12. Назначение монады типа `Maybe`.
13. Дать определение ленивого интерпретатора.

Примерный перечень курсовой работы

1. Разработка игры в «крестики-нолики» с переменным размером поля.
2. Разработка программы трансформации XML документов. Пример: вывод статистики о сценарии фильма и преобразование `xml` в `html`.
3. Разработка программы поиска текстовых подстрок по заданному регулярному выражению по файлу или группе файлов.
4. Разработка программы поиска кратчайшего пути в лабиринте.
5. Разработка программы-калькулятора с функциональностью инженерного калькулятора.
6. Разработка справочно-информационной системы авиакомпании. Система должна позволять работать с базой рейсов и подбирать рейсы (с учетом стыковок рейсов в течение одних суток) по минимальной стоимости билетов, минимальному времени в пути.
7. Разработка информационной системы транспорта города Владимира.

8. Разработка информационной системы оптимизации перевозки грузов.
9. Разработка автоматизированной системы планирования встреч и совещаний.
10. Разработка экспертной системы по классификации объектов некоторой предметной области. Предметная область – комнатные растения – используется для тестирования. Так же выбрать пару предметных областей на свое усмотрение и подготовить файлы для тестирования программы для них.
11. Разработка экспертной системы по подбору подходящего требованиям клиента телефона.
12. Разработка экспертной системы по подбору подходящего требованиям клиента тарифа сотового оператора.
13. Разработка информационной системы туристической фирмы. В программе должна быть возможность формирования базы туров и подбора туров по критериям.
14. Разработка экспертной системы по ремонту/настройке устройств некоторой предметной области. (Например, устранение некоторых неполадок ОС)
15. Разработка программы поиска файлов на диске. В программе предусмотреть возможность интерактивного задания условий поиска. Реализовать с применением графического интерфейса.
16. Разработка справочно-информационной системы железнодорожных перевозок. Система должна позволять работать с базой рейсов и подбирать рейсы (с учетом стыковок рейсов в течение одних суток) по минимальной стоимости билетов, минимальному времени в пути.
17. Разработка программы расчета выдачи наличных в банкоматах.
18. Реализация игры «Быки и коровы».
19. Разработка программы раскраски карты. В программе учесть требования, что две соседние страны не могут быть закрашены одинаково.
20. Разработка модуля поиска по библиотечной базе данных. В работе должно быть реализовано хранение анализируемой информации, поиск публикаций должен осуществляться по названиям, автору, издательству, ключевым словам.
21. Разработка программы, реализующей выполнение подмножества команд языка SQL. В программе реализовать базу данных, заполнить ее, применить запросы на созданной БД.
22. Разработка системы символьного дифференцирования, интегрирования и упрощения некоторых типов выражений.
23. Система поиска маршрута по сети линий метро. Информация должна быть представлена наглядно в графическом виде.
24. Разработка программы обработки результатов социологических опросов. Программа должна иметь функционал для сбора сведений, анализа и обобщения результатов социологических опросов.
25. Разработка программы обработки математических выражений. Программа должна уметь представлять математические выражения в виде обратной польской записи и вычислять выражения по их обратной польской записи.
26. Разработка справочно-информационной системы авиаперевозок. Система должна позволять работать с базой рейсов и подбирать рейсы (с учетом стыковок рейсов в течение одних суток) по минимальной стоимости билетов, минимальному времени в пути.

Самостоятельная работа студента (контроль освоения всех разделов курса)

Следует отметить, что в связи с существенной практической направленностью курса, существенный вес имеют результаты, полученные студентами в рамках практической самостоятельной работы.

На самостоятельную работу выносятся проработка каждого лекционного занятия по соответствующим темам и разделам дисциплины. Доработка лабораторных работ, оформление отчетов по лабораторным работам и подготовка к защите лабораторных работ.

Для контроля усвоения материала, изученного в рамках самостоятельной работы студент должен знать ответы на контрольные вопросы к каждому разделу дисциплины. Приведенные контрольные вопросы можно применять при защите лабораторных работ, курсовой работы в рамках изученных разделов.

Контрольные вопросы:

Раздел 1. Введение в функциональное программирование

1. Какие языки функционального программирования знаете?
2. Перечислите основные задачи, для которых наиболее удобно применить функциональное программирование в качестве средства решения.
3. В чем заключается принцип чистоты функциональной программы?
4. Как понимаете выражение «функции - значения»?

Раздел 2. Лямбда-исчисление Чёрча.

1. Формальная нотация лямбда-терма.
2. Форма Бэкуса-Наура.
3. Свободные и связанные переменные. В чем особенности работы с такими переменными?
4. Определить понятие равенства лямбда-термов.
5. Редукционные стратегии.
6. Как можно представить натуральные числа в форме лямбда-термов и выполнить элементарные математические операции?

7. Применение именованных выражений. Приведите примеры.

8. Особенности типизированного лямбда-исчисления.

Раздел 3. Принципы функционального программирования.

1. В чем отличие команд интерпретатора от выражений языка Haskell?
2. Основные типы языка Haskell.
3. Функции для работы с кортежами.
4. Функции для работы со списками.
5. Допустимые имена переменных и функций.
6. Команды интерпретатора для работы с файлами программ.
7. Условные выражения в языке Haskell.
8. Определение функций в языке Haskell.

Раздел 4. Языки функционального программирования.

1. Правила выравнивания.
2. Сопоставление с образцом.
3. Операция выбора.
4. Кусочное задание функций.
5. Определение локальных переменных.
6. Охраняющие условия.

7. Полиморфизм.

8. Определение пользовательских типов данных.

Раздел 5. Интерпретация языков функционального программирования.

1. Монады. Определение, назначение и примеры применения.

2. Что такое функция высшего порядка и порядок работы с такими функциями.

3. Реализация многопоточности в функциональной программе.

4. Дать определение ленивого интерпретатора.

5. Дать определение энергичного интерпретатора

Раздел 6. Применения функционального программирования. Перспективы развития

1. Расскажите о внедрении особенностей функционального программирования в объектно-ориентированное программирование.

2. Перечислите сферы применения языков функционального программирования.

3. Применение для решения задач доказательства теорем.

4. Особенности обработки рекурсивных кодов интерпретаторами функционального языка.

Экзамен (контроль освоения всех разделов курса)

Предполагает ответ студента на два теоретических вопроса. Теоретические вопросы сгруппированы в комплект билетов, который формируется на основе включения вопросов из различных разделов дисциплины.

Ответы на теоретические вопросы должны быть подкреплены практическими примерами, написанными на одном из функциональных языков.

Базовый список теоретических вопросов:

1. Понятие о функциональном программировании. Основные принципы ФП.
2. Сравнение ФП с императивным программированием.
3. Применение функционального программирования для решения задач.
4. ФП и системы искусственного интеллекта.
5. Классификация лямбда-термов. Форма Бэкуса-Наура.
6. Свободные и связанные переменные.
7. Конверсия и равенство лямбда-выражений.
8. Редукция лямбда-выражений и редукционные стратегии.
9. Комбинаторы. Комбинатор неподвижной точки.
10. Представление данных в лямбда-исчислении: булевские значения, условия, пары и кортежи.
11. Представление данных в лямбда-исчислении: натуральные числа и операции над ними.
12. Именованные выражения `let` и `where`.
13. Типизированное лямбда-исчисление. Типизация по Черчу и Карри.
14. Нормальная форма лямбда-выражений. Сильная нормализация.
15. Рекурсии в лямбда-исчислении.
16. Функции высших порядков.
17. Рекурсивные функции.
18. Статическое и динамическое связывание.
19. Отложенные вычисления.
20. Карринг.
21. Базовые типы языка Haskell.

22. Списки в языке Haskell.
23. Кортежи в языке Haskell.
24. Охранные выражения.
25. Сопоставление с образцом. Пример.
26. Определение рекурсивных функций. Примеры.
27. Операторы языка Haskell. Префиксная и инфиксная нотации. Приоритеты. Ассоциативность.
28. Карринг в языке Haskell.
29. Функции высшего порядка: Функции на списках.
30. Функции высшего порядка: Лямбда функции.
31. Функции высшего порядка: Композиция функций.
32. Бесконечные списки. Принцип построения. Примеры.
33. Объектно-ориентированные возможности языка Haskell.
34. Встроенные классы.
35. Определение новых типов данных в языке Haskell.
36. Монады.
37. Ввод и вывод информации в языке Haskell. Файлы.
38. SECD-машина.
39. Ленивый и энергичный интерпретатор.
40. Метод редукции графов.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. Городняя, Л. В. Основы функционального программирования [Электронный ресурс] / Л. В. Городняя. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 246 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73703.html>
2. Душкин, Р. В. Функциональное программирование на языке Haskell [Электронный ресурс] / Р. В. Душкин. — Электрон. текстовые данные. — Саратов : Профобразование, 2017. — 608 с. — 978-5-4488-0044-3. — Режим доступа: <http://www.iprbookshop.ru/64053.html>
3. Салмина, Н. Ю. Функциональное программирование и интеллектуальные системы [Электронный ресурс] : учебное пособие / Н. Ю. Салмина. — Электрон. текстовые данные. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2016. — 100 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/72216.html>

б) дополнительная литература:

1. Практикум по дисциплине Логическое и функциональное программирование [Электронный ресурс] / сост. А. С. Доткулова. — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 16 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61489.html>
2. Учебно-методическое пособие по дисциплине Логическое и функциональное программирование [Электронный ресурс] / сост. М. В. Яшина, В. В. Барков, С. В. Украинский. — Электрон. текстовые данные. — М. : Московский

технический университет связи и информатики, 2016. — 23 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61490.html>

3. Сошников, Д. В. Функциональное программирование на F# [Электронный ресурс] / Д. В. Сошников. — Электрон. текстовые данные. — Саратов : Профобразование, 2017. — 191 с. — 978-5-4488-0131-0. — Режим доступа: <http://www.iprbookshop.ru/63961.html>

4. Сузи, Р. А. Язык программирования Python [Электронный ресурс] / Р. А. Сузи. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 350 с. — 5-9556-0058-2. — Режим доступа: <http://www.iprbookshop.ru/52211.html>

5. Уэс, Маккинли Python и анализ данных [Электронный ресурс] / Маккинли Уэс ; пер. А. А. Слинкин. — Электрон. текстовые данные. — Саратов : Профобразование, 2017. — 482 с. — 978-5-4488-0046-7. — Режим доступа: <http://www.iprbookshop.ru/64058.html>

6. Буйначев, С. К. Основы программирования на языке Python [Электронный ресурс] : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг ; под ред. Ю. В. Песин. — Электрон. текстовые данные. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. — 92 с. — 978-5-7996-1198-9. — Режим доступа: <http://www.iprbookshop.ru/66183.html>

в) Интернет-ресурсы:

1. Haskell -- a purely functional language [Электронный ресурс]. <http://www.haskell.org/haskellwiki/Haskell>.
2. The Concurrent Clean Language Report [Электронный ресурс]. <http://www.cs.ru.nl/~clean/contents/contents.html>.


8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

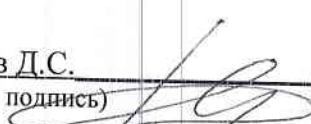
Учебный сервер во внутренней сети университета, на котором функционирует сайт дисциплины. Кроме того, на сервере студенты размещают результаты выполнению лабораторных и курсовых работ.


Комплект мультимедийных презентаций по 5-му разделу курса. В случае отсутствия возможности проведения занятий в уже оборудованной мультимедийным проектором аудитории используются переносной экран и проектор, имеющиеся на кафедре ФИИМ.

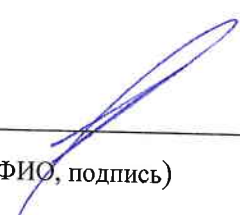
Среда разработки Microsoft Visual Studio, GHC, Haskell Platform.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем

Рабочую программу составил ст. преподаватель каф. ФиПМ Павлова О.Н.
(ФИО, подпись) 

Рецензент
(представитель работодателя) Ген. директор ООО «ФС Сервис» Квасов Д.С.
(место работы, должность, ФИО, подпись) 

Программа рассмотрена и одобрена на заседании кафедры ФиПМ
Протокол №1 от 03.09.2018 года
Заведующий кафедрой _____ Аракелян С.М.
(ФИО, подпись) 

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.03 Математическое обеспечение и администрирование информационных систем
Протокол №1 от 03.09.2018 года
Председатель комиссии _____ Аракелян С.М.
(ФИО, подпись) 

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____