

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра «Программного обеспечения автоматизированных систем»



УТВЕРЖДАЮ:
Ректор

Н.В. Дубив

2020 г.

Рабочая программа

МОДУЛЯ

ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

образовательной программы высшего образования –
программы магистратуры

09.04.04 Программная инженерия
направленность

*Методы и алгоритмы интеллектуальной обработки данных
в информационно-вычислительных системах*

формы обучения – очная

Рабочая программа составлена в соответствии с учебными планами программы магистратуры Программная инженерия (Методы и алгоритмы интеллектуальной обработки данных в информационно-вычислительных системах) очной формы обучения, утвержденными 28.08.2020 г.

Рабочая программа одобрена на заседании кафедры Программного обеспечения автоматизированных систем 31.08.2020 года, протокол № 1.

Рабочую программу разработали:

зав. кафедрой ПОАС



Т.Р. Змызгова

доцент кафедры ПОАС



А.А. Медведев

ст. преп. кафедры ПОАС



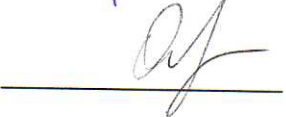
Д.А. Подкорытов

доцент кафедры ПОАС



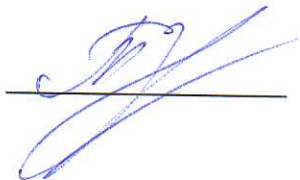
А.Б. Флока

доцент кафедры ПОАС



О.С. Черепанов

Заведующий
кафедрой ПОАС



Т.Р. Змызгова

Согласовано:

Начальник
Управления
образовательной деятельности



С.Н. Сеницын

Специалист
по учебно-методической работе
Учебно-методического отдела



Г.В. Казанкова

1. ОБЪЕМ МОДУЛЯ

Модуль «Языки и технологии программирования» содержит 12 элективных дисциплин, 6 из которых (по три дисциплины в первом и втором семестрах) должны быть выбраны студентами для изучения:

| <u>1-й семестр:</u> | <u>2-й семестр:</u> |
|---------------------|---------------------|
| Язык Python | Язык Julia |
| Язык C++ | Язык C# |
| Язык Java | Язык Kotlin |
| Язык JavaScript | Язык Go |
| Язык Haskell | Язык Prolog |
| Язык Erlang | Язык Ocaml |

Все 12 дисциплин идентичны по трудоемкости (2 зач. ед. – 72 акад. часа) и структуре учебных занятий:

| Виды учебной работы | Распределение трудоемкости одной дисциплины модуля по видам учебных занятий (акад. часов) |
|--------------------------|---|
| Аудиторные занятия: | 24 |
| Лекции | 8 |
| Лабораторные занятия | 16 |
| Самостоятельная работа: | 48 |
| Подготовка к зачету | 18 |
| Прочие виды | 30 |
| Промежуточная аттестация | Зачет |

Общая трудоемкость модуля – 12 зач. ед. (432 акад. часа)

| Виды учебной работы | Распределение трудоемкости по семестрам и видам учебных занятий (акад. часов) | | |
|--------------------------|---|-------------|-------------|
| | Всего | 1-й семестр | 2-й семестр |
| Трудоемкость, з.е. | 12 | 6 | 6 |
| Аудиторные занятия: | 144 | 72 | 72 |
| Лекции | 48 | 24 | 24 |
| Лабораторные занятия | 96 | 48 | 48 |
| Самостоятельная работа: | 288 | 144 | 144 |
| Подготовка к зачету | 108 | 54 | 54 |
| Прочие виды | 180 | 90 | 90 |
| Промежуточная аттестация | | 3 зачета | 3 зачета |

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Модуль «Языки и технологии программирования» включен в блок элективных дисциплин вариативной части учебного плана (дисциплины по выбору).

Для освоения дисциплин модуля необходимы компетенции в области компьютерного программирования, формируемые соответствующими дисциплинами программ бакалавриата или специалитета.

Результаты изучения дисциплин модуля используются при выполнении лабораторных и практических заданий по дисциплинам учебного плана, а также при выполнении междисциплинарных курсовых работ и выпускной квалификационной работы.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплин модуля – формирование теоретических знаний и практических навыков разработки программного обеспечения.

Задачи дисциплин:

изучение:

- парадигм структурного, объектно-ориентированного и функционального программирования и их реализаций в различных языках;
- основных типов данных и операций над ними;

практическое освоение:

- языков программирования, реализующие различных парадигмы.
- этапов реализации программного обеспечения.

Компетенции, формируемые в результате освоения дисциплин:

- способность разрабатывать и использовать программное обеспечение для моделирования, анализа, распознавания и обработки информации, в том числе – в системах искусственного интеллекта (ПК-3).

В результате освоения дисциплины студент должен демонстрировать следующие **результаты обучения**:

Должен знать:

- парадигмы программирования и технологии их реализации;
- основные типы данных и операции над ними;
- назначение и правила оформления основных конструкций языка.

Должен уметь:

- выбирать парадигму программирования, язык, библиотеки и другие программные средства на основе результатов анализа условий задачи;
- выполнять проектирование, кодирование, отладку, рефакторинг и тестирование разрабатываемого ПО с использованием соответствующих программных сред.

Должен владеть:

- терминологией в области реализации программного обеспечения;
- технологиями реализации парадигм программирования средствами выбранного языка программирования;
- навыками реализации (проектирования, кодирования, отладки, рефакторинга, тестирования) программного обеспечения.

4. СОДЕРЖАНИЕ МОДУЛЯ

4.1 Дисциплина «Язык C++»

4.1.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|---|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы языка C++ | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Библиотеки для решения задач машинного обучения | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.1.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка C++ | |
| Лекция 1. Типы данных. Операторы. Функции. Переменные и базовые типы. Основные операции над типами данных. Типы STL. Итераторы. Операторы: условные операторы, итерационные операторы, операторы перехода. Функции. Передача аргументов. Типы возвращаемого значения и оператор return. Перегрузка функций. | 2 |
| Лекция 2. Классы. Наследование. Шаблоны. Области видимости. Пространство имен. Классы. Свойства. Конструкторы. Деструкторы. Методы. Указатель this. Управление доступом. Виртуальные методы. Статические члены класса. Наследование. Модификаторы доступа при наследовании. Множественное наследование. Ромбовидное наследование. Обработка исключений. Шаблоны. | 2 |
| Раздел №2. Библиотеки для решения задач машинного обучения | |
| Лекция 3. <i>mlpack</i>. Решение прикладных задач, используя библиотеку <i>mlpack</i> , путем сведения их к композиции типовых задач. Импорт данных. Пакет Matplotlib-cpp. Предварительный анализ. Решение задач классификации, регрессии, кластерного анализа, прогнозирования. | 4 |
| Всего часов лекционных занятий | 8 |

4.1.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка C++ | |
| Лабораторная работа № 1. Проектирование и реализация программного обеспечения, включающего основные отношения между классами и объектами. Тема согласуется с преподавателем. | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. Библиотеки решения задач машинного обучения | |
| Лабораторная работа № 2. Использование библиотек машинного обучения для решение прикладной задачи на согласованную с преподавателем тему. | 8 |
| Рубежный контроль №2. | 2 |

| | |
|--------------------------------|----|
| Всего часов лабораторных работ | 16 |
|--------------------------------|----|

4.2 Дисциплина «Язык С#»

4.2.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|----------------------|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы языка С# | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | GraphQL | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.2.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|--|-------------------------|
| Раздел №1. Основы языка С# | |
| Лекция 1. Типы данных. Операторы. Введение в язык С#. Переменные и типы данных. Массивы и строки. Перечисления. Преобразование типов. Операторы. Управляющие операторы: условный оператор, оператор выбора, итерационные операторы, оператор перехода. | 2 |
| Лекция 2. Классы. Наследование. Интерфейсы. Классы. Поля. Методы. Передача аргументов в методы. Возврат значений из методов. Конструкторы. Параметризованные конструкторы. Деструкторы. Ключевое слово this. Управление доступом к членам класса. Перегрузка методов и конструкторов. Применение ключевого слова static. Перегрузка операторов. Индексаторы. Наследование. Виртуальные методы. Абстрактные классы. Интерфейсы. Наследование интерфейсов. Обработка исключительных ситуаций. Делегаты, события и лямбда-выражения. | 2 |
| Раздел №2. GraphQL | |
| Лекция 3. Основы GraphQL. Назначение и функционал GraphQL. Сравнение с REST. Поля. Аргументы. Фрагменты. Переменные. Директивы. Мутации. Схемы. Система типов. Типы объектов и поля. Запросы. Скалярные типы. Перечисления. Списки. Интерфейсы. Объединения. Типы ввода. Подписчики. Валидация запросов. Resolvers. | 2 |
| Лекция 4. Реализация GraphQL API сервера. Apollo Server. Создание нового проекта. Установка зависимостей. Определение GraphQL-схемы. Определение Resolver. Разворачивание приложения. Тестирование. | 2 |
| Всего часов лекционных занятий | 8 |

4.2.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка С# | |
| Лабораторная работа № 1. Проектирование и реализация программного обеспечения, включающего основные отношения между классами и объектами. Тема согласуется с преподавателем. | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. GraphQL | |

| | |
|--|----|
| Лабораторная работа № 2. Реализация GraphQL API сервера. | 8 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.3 Дисциплина «Язык Julia»

4.3.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|---|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы языка Julia | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Библиотеки для решения задач машинного обучения | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.3.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка Julia | |
| Лекция 1. Типы данных. Операции над типами данных. Функции. Управляющие конструкции языка. Введение в язык Julia, основные особенности языка. Переменные. Базовые типы данных. Основные операции с типами данных. Функции. Аргументы функции. Оператор возврата. Анонимные функции. Композиция функций. Управляющие конструкции языка. Составные выражения. Условные операторы. Тернарный оператор. Циклы while, for. Корутины. | 2 |
| Лекция 2. Составные типы. Методы и конструкторы. Интерфейсы. Модули. Область видимости переменных. Локальные и глобальные переменные. Let-блок. Системы типов. Абстрактные типы. Составные типы. Мутабельные составные типы. Объединения. Параметрические типы. Параметрические абстрактные типы. Одиночки. Операции над типами. Методы. Определение методов. Параметрические методы. Переопределение методов. Конструкторы. Параметрические конструкторы. | 2 |
| Раздел №2. Библиотеки для решения задач машинного обучения | |
| Лекция 3. JuliaStats и Flux. Решение прикладных задач, используя пакеты JuliaStats и Flux, путем сведения их к композиции типовых задач. Импорт данных. Пакет Plots. Предварительный анализ. Задачи классификации, регрессии, кластерный анализ, прогнозирование. | 4 |
| Всего часов лекционных занятий | 8 |

4.3.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ ЯЗЫКА JULIA | |
| Лабораторная работа №1. Проектирование и реализация программного обеспечения, включающего основные отношения между классами и объектами. Тема согласуется с преподавателем. | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. Библиотеки решения задач машинного обучения | |

| | |
|--|----|
| Лабораторная работа №2. Использование библиотек машинного обучения для решение прикладной задачи на согласованную с преподавателем тему. | 8 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.4 Дисциплина «Язык Python»

4.4.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|---|--|------------------|
| № | Наименование | | |
| | | Лекции | Лаборат. занятия |
| 1 | Особенности языка и реализации Python-приложений | 4 | 6 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Обзор Python-библиотек для работы с большими массивами данных | 4 | 6 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.4.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. ОСОБЕННОСТИ ЯЗЫКА И РЕАЛИЗАЦИИ PYTHON-ПРИЛОЖЕНИЙ | |
| Лекция 1. Сравнительная характеристика и особенности использования языка программирования Python. Краткая характеристика языка. Особенности использования изменяемых и неизменяемых типов переменных. Циклы и итерации. Выражения-генераторы. Форматирование строковых значений. Функции и декораторы. Аргументы функции. Итерации и менеджеры контекста | 2 |
| Лекция 2. Особенности реализации классов и использования объектов в Python. Сравнение объектов. Классы-исключения. Наследование. Переменные и методы класса и экземпляра. Особенности реализации методов класса. Основы библиотеки PyQt5 | 2 |
| Раздел №2. ОБЗОР PYTHON-БИБЛИОТЕК ДЛЯ РАБОТЫ С БОЛЬШИМИ МАССИВАМИ ДАННЫХ | |
| Лекция 3. Обзор библиотек, используемых для хранения больших массивов данных и работы с ними. Библиотека NumPy: стандартные типы данных, работа с массивами. Транслирование, сравнения и маски. Сортировка массивов. Библиотека Pandas: объекты Series, DataFrame, Index. Обработка отсутствующих данных. Объединение наборов данных | 2 |
| Лекция 4. Двумерная и трехмерная визуализация данных в Python. Библиотека Matplotlib: простые линейные графики, простые диаграммы рассеяния, визуализация погрешностей. Гистограммы. Настройки графиков. Трехмерные графики. Библиотека VisPy и/или MayaVi. | 2 |
| Всего часов лекционных занятий | 8 |

4.4.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСОБЕННОСТИ ЯЗЫКА И РЕАЛИЗАЦИИ PYTHON-ПРИЛОЖЕНИЙ | |
| Лабораторная работа № 1. Реализация простых Python-приложений. Тема согласуется с преподавателем | 6 |
| Рубежный контроль №1. | 2 |
| Раздел №2. ОБЗОР PYTHON-БИБЛИОТЕК ДЛЯ РАБОТЫ С БОЛЬШИМИ МАССИВАМИ ДАННЫХ | |

| | |
|---|----|
| Лабораторная работа № 2. Создание приложения по обработке больших наборов данных и визуализации полученных результатов на согласованную с преподавателем тему | 6 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.5 Дисциплина «Язык Kotlin»

4.5.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|--|--|------------------|
| № | Наименование | Лекции | Лаборат. Занятия |
| | | 1 | Основы Kotlin |
| | Рубежный контроль №1 | - | 2 |
| 2 | Объектно-ориентированная и функциональная парадигмы в Kotlin | 4 | 6 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.5.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ KOTLIN | |
| Лекция 1. Особенности синтаксиса и использования языка программирования Kotlin. Назначение и характеристика языка. Основные типы. Объявления переменных. Условные и циклические конструкции. Строки и коллекции. Функции | 2 |
| Лекция 2. Стандартные функции и структуры данных. Функции apply, let, run, with also и др. Списки и множества. Ассоциативные массивы | 2 |
| Раздел №2. ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ И ФУНКЦИОНАЛЬНАЯ ПАРАДИГМЫ В KOTLIN | |
| Лекция 3. Классы, объекты и интерфейсы. Особенности определения классов и интерфейсов. Лямбда-выражения. Переменные и методы класса и экземпляра. Особенности реализации методов класса. Объявление объектов. Особенности ООП. Анонимные и вспомогательные объекты. Обобщения и расширения. | 2 |
| Лекция 4. Основы функционального программирования. Категории функций: преобразователи, фильтры и комбинаторы. Последовательности. Библиотека Arrow.kt, ее основные функции | 2 |
| Всего часов лекционных занятий | 8 |

4.5.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ KOTLIN | |
| Лабораторная работа № 1. Реализация простых Kotlin-приложений. Тема согласуется с преподавателем | 6 |
| Рубежный контроль №1. | 2 |
| Раздел №2. ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ И ФУНКЦИОНАЛЬНАЯ ПАРАДИГМЫ В KOTLIN | |
| Лабораторная работа № 2. Разработка и использование классов в Kotlin-приложениях по согласованной с преподавателем теме | 3 |

| | |
|---|----|
| Лабораторная работа № 3. Использование элементов функционального программирования при разработке Kotlin-приложений по согласованной с преподавателем теме | 3 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.6 Дисциплина «Язык Haskell»

4.6.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|--|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы Haskell | 4 | 6 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Списки в Haskell | 2 | 3 |
| 3 | Конкурентное и параллельное программирование в Haskell | 2 | 3 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.6.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. ОСНОВЫ HASKELL | |
| <i>Лекция 1. Особенности синтаксиса и использования языка программирования Haskell.</i> Назначение и характеристика языка. Основные типы и операции. Определение функций. Условные конструкции. Основные функции | 2 |
| <i>Лекция 2. Рекурсия в Haskell.</i> Простая рекурсия на числовых структурах (по аргументам, по значению (хвостовая рекурсия), по значению и по аргументам), нисходящая и восходящая рекурсия. Сложные типы рекурсии на числовых структурах | 2 |
| Раздел №2. СПИСКИ В HASKELL | |
| <i>Лекция 3. Списки в Haskell.</i> Список как фундаментальный тип данных. Конструирование списков. Операции над списками. Функции обработки списков. Рекурсия на списках. Поиск в списках и сортировка списков | 2 |
| Раздел №3. КОНКУРЕНТНОЕ И ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ В HASKELL | |
| <i>Лекция 4. Обзор основных библиотек для реализации параллелизма и конкурентного кода.</i> Понятие монады. Монады Eval, Par, их использование для реализации параллелизма. Обзор библиотеки Rera. Использование базовых интерфейсов для написания конкурентного кода | 2 |
| Всего часов лекционных занятий | 8 |

4.6.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ HASKELL | |
| Лабораторная работа № 1. Реализация простых Haskell-приложений. Тема согласуется с преподавателем | 6 |
| Рубежный контроль №1. | 2 |
| Раздел №2. СПИСКИ В HASKELL | |

| | |
|---|-----------|
| Лабораторная работа № 2. Разработка приложений на обработку списков по теме, согласованной с преподавателем | 3 |
| Раздел №2. КОНКУРЕНТНОЕ И ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ В HASKELL | |
| Лабораторная работа № 3. Создание простейших приложений с использованием элементов конкурентного и/или параллельного программирования. | 3 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.7 Дисциплина «Язык Java»

4.7.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|-----------------------|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы языка Java | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Java фреймворк Spring | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.7.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка Java | |
| Лекция 1. Типы данных. Операторы. Управляющие инструкции. Примитивные типы данных. Инициализация переменных. Области действия переменных в методе. Арифметические операторы. Операторы сравнения и логические операторы. Операторы присваивания. Преобразование типов при присваивании. Приведение несовместимых типов данных. Полная форма условной инструкции if. Инструкция switch. Полная форма цикла for. Цикл while. Цикл do-while. Инструкции break, continue. Вложенные циклы. | 2 |
| Лекция 2. Классы. Объекты. Методы. Наследование. Пакеты и интерфейсы. Модули. Классы, создание объектов класса. Ссылки на объекты. Создание методов. Применение конструкторов. Ключевые слова new и this. Сборка мусора. Создание многоуровневой иерархии классов. Методика переопределения метода. Абстрактные классы. Ключевое слово final. Класс Object. Использование пакетов. Стандартные пакеты Java. Интерфейсы, реализация интерфейсов. Объявление модуля с помощью ключевого слова module. Инструкции requires и exports. Назначение файла module-info.java. Назначение модуля java.base. Экспорт пакета для указанного модуля. | 2 |
| Раздел №2. Java фреймворк Spring | |
| Лекция 3. Инверсия управления (IoC) и внедрение зависимостей (DI) в Spring. Аспектно-ориентированное программирование (АОП). Работа с базами данных, JDBC, JPA. Разработка web-приложений Spring MVC. Принципы инверсии управления. Инверсия управления в Spring. Внедрение зависимостей в Spring. Конфигурирование контекста приложений. Принципы АОП. Архитектура АОП в Spring. Реализация JDBC в Spring. Подключение к базе данных, извлечение и преобразование данных в Spring. Библиотека объектно-реляционного преобразования (ORM). Прикладной интерфейс сохраняемости Java persistence API (JPA). Модуль Spring MVC. | 4 |
| Всего часов лекционных занятий | 8 |

4.7.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
|---|-------------------------|

| Раздел №1. Основы языка Java | |
|--|----|
| Лабораторная работа № 1. Проектирование и реализация программного обеспечения, включающего основные отношения между классами и объектами. Тема согласуется с преподавателем. | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. Java фреймворк Spring | |
| Лабораторная работа № 2. Проектирование и реализация web-приложения Spring, с использованием Spring MVC, Spring JDBC, JPA. | 8 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.8 Дисциплина «Язык JavaScript»

4.8.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|-------------------------|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы языка Javascript | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Javascript фреймворки | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.8.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|--|-------------------------|
| Раздел №1. Основы языка Javascript | |
| Лекция 1. <i>Типы данных. Операторы. Функции.</i> Структура кода. Переменные. Типы данных. Преобразование типов. Операторы. Операторы сравнения. Взаимодействие: alert, prompt, confirm. Условные операторы if, '?'. Логические операторы. Циклы while, for. Конструкция switch. Функции. Функции-стрелки. | 2 |
| Лекция 2. <i>Классы. Объекты. Типы данных. Продвинутая работа с функциями.</i> Классы в JavaScript. Наследование классов. Статические свойства и методы. Приватные и защищенные методы и свойства. Расширение встроенных классов. Проверка класса instanceof. Объекты, метод this. Конструкторы, создание объектов через new. Числа, строки, массивы. Map и Set. Формат JSON, метод toJSON. Замыкание. Синтаксис new Function. Привязка контекста функции. | 2 |
| Раздел №2. Javascript фреймворки | |
| Лекция 3. React. Vue. Angular. Альтернатива JS Фреймворкам - Google flutter. Особенности использования и применения фреймворков React. Vue. Angular. Преимущества и недостатки фреймворков React. Vue. Angular. Базовые наборы библиотек фреймворков React. Vue. Angular. Новые подход к проектированию и созданию frontend приложений Google flutter. | 4 |
| Всего часов лекционных занятий | 8 |

4.8.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка Javascript | |
| Лабораторная работа № 1. Проектирование и реализация про- | 4 |

| | |
|--|----|
| граммного обеспечения javascript, включающего основные отношения между классами и объектами. Тема согласуется с преподавателем. | |
| Рубежный контроль №1. | 2 |
| Раздел №2. Javascript фреймворки | |
| Лабораторная работа № 2. Использование фреймворков React. Vue. Angular, Google flutter для решение прикладной задачи на согласованную с преподавателем тему. | 8 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.9 Дисциплина «Язык Prolog»

4.9.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|---------------------------------|--|-----------------|
| № | Наименование | | |
| | | Лекции | Лабора. занятия |
| 1 | Основы языка Prolog | 4 | 4 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Основы теории экспертных систем | 4 | 8 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.9.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|--|-------------------------|
| Раздел №1. Основы языка Prolog | |
| Лекция 1. Основы языка Prolog. Пролог и технология программирования. Общая характеристика языка. Основные понятия языка Пролог. Структура программы. Стандартные домены. Логические основы языка Пролог (предикат, факт, правило). Предикаты, утверждения, правила. Процедурный подход. Управление вычислениями. Рекурсия. Методы организации рекурсии: простая рекурсия, метод обобщенного правила рекурсии. Строки и списки. Пролог в системах искусственного интеллекта. Пролог и базы данных. Встроенные предикаты Пролога для написания базы данных. | 4 |
| Раздел №2. Основы теории экспертных систем | |
| Лекция 3. Проблематика и технологии экспертных систем. Назначение экспертных систем и области их применения. Технология разработки простейшей экспертной системы. Структура и области применения. Методы, используемые при разработке экспертных систем: прямая цепочка рассуждений, обратная цепочка рассуждений. Требования к системам знаний. Виды экспертных систем. Разработка простейших экспертных систем. | 4 |
| Всего часов лекционных занятий | 8 |

4.9.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Основы языка Prolog | |

| | |
|---|----|
| Лабораторная работа № 1. Логическое программирование в среде Prolog. Построение базы данных или базы знаний по теме, согласованной с преподавателем. | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. Основы теории экспертных систем | |
| Лабораторная работа № 2. Создание программной экспертной системы для предметной области, согласованной с преподавателем. | 8 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

4.10 Дисциплина «Язык Erlang»

4.10.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|--|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Основы Erlang. Парадигмы и синтаксис. | 4 | 6 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Основы распределенных систем на Erlang OTP | 4 | 6 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.10.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ ERLANG. ПАРАДИГМЫ И СИНТАКСИС | |
| Лекция 1. Основы и парадигма языка Erlang | 2 |
| Лекция 2. Синтаксис и примеры кода Erlang | 2 |
| Раздел №2. ОСНОВЫ РАСПРЕДЕЛЕННЫХ СИСТЕМ НА ERLANG OTP | |
| Лекция 3. Распределенная модель вычислений в Erlang OTP | 2 |
| Лекция 4. Мониторы и супервизоры | 2 |
| Всего часов лекционных занятий | 8 |

4.10.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|--|-------------------------|
| Раздел №1. ОСНОВЫ ERLANG. ПАРАДИГМЫ И СИНТАКСИС. | |
| Лабораторная работа №1. Обработка списков. Циклы в Erlang | 2 |
| Лабораторная работа №2. Обработка деревьев в Erlang | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. ОСНОВЫ РАСПРЕДЕЛЕННЫХ СИСТЕМ НА ERLANG OTP | |
| Лабораторная работа №3. Разработка микросервиса на Erlang | 2 |
| Лабораторная работа №4. Разработка простейшей кластерной системы с супервизором | 4 |
| Рубежный контроль №1. | 2 |

| | |
|--------------------------------|----|
| Всего часов лабораторных работ | 16 |
|--------------------------------|----|

4.11 Дисциплина «Язык Go»

4.11.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|---|--|------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| 1 | Язык Golang , его парадигма и синтаксис | 4 | 6 |
| | Рубежный контроль №1 | - | 2 |
| 2 | Взаимодействие процессов в Golang | 4 | 6 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.11.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Язык GOLANG , ЕГО ПАРАДИГМА И СИНТАКСИС | |
| Лекция 1. Синтаксис Golang и основные принципы языка | 2 |
| Лекция 2. Типовые примеры кода | 2 |
| Раздел №2. ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОВ В GOLANG | |
| Лекция 3. Горутины | 2 |
| Лекция 4. Каналы | 2 |
| Всего часов лекционных занятий | 8 |

4.11.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. Язык GOLANG , ЕГО ПАРАДИГМА И СИНТАКСИС | |
| Лабораторная работа №1. Обработка массива в Golang | 2 |
| Лабораторная работа №2. Обработка списка в Golang | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОВ В GOLANG | |
| Лабораторная работа №3. Пример с горутиной | 2 |
| Лабораторная работа №4. Микросервис на горутинах и каналах | 4 |
| Рубежный контроль №1. | 2 |
| Всего часов лабораторных работ | 16 |

4.12 Дисциплина «Язык OCaml»

4.12.1 Учебно-тематический план

| Разделы дисциплины | | Часов контактной работы с преподавателем | |
|----------------------|--------------------------|--|------------------------------|
| № | Наименование | Лекции | Лаборат. занятия |
| | | 1 | OCaml: парадигма и синтаксис |
| | Рубежный контроль №1 | - | 2 |
| 2 | Разработка кода на OCaml | 4 | 6 |
| | Рубежный контроль №2 | - | 2 |
| Всего по дисциплине: | | 8 | 16 |

4.12.2 Содержание лекционных занятий

| Наименование и содержание лекции | Часов контактной работы |
|---|-------------------------|
| Раздел №1. ОСАМЛ: ПАРАДИГМА И СИНТАКСИС | |
| Лекция 1. Парадигмы OCaml | 2 |
| Лекция 2. Синтаксис OCaml | 2 |
| Раздел №2. РАЗРАБОТКА КОДА НА ОСАМЛ | |
| Лекция 3. Обработка списков и деревьев в ОСАМЛ | 2 |
| Лекция 4. Toolchain для выполнения проектов на ОСАМЛ | 2 |
| Всего часов лекционных занятий | 8 |

4.12.3 Лабораторные работы

| Наименование и содержание лабораторной работы | Часов контактной работы |
|---|-------------------------|
| Раздел №1. ОСАМЛ: ПАРАДИГМА И СИНТАКСИС | |
| Лабораторная работа № 1. Использование функций и переменных при написании кода ОСАМЛ | 2 |
| Лабораторная работа № 2. Обработка списков | 4 |
| Рубежный контроль №1. | 2 |
| Раздел №2. РАЗРАБОТКА КОДА НА ОСАМЛ | |
| Лабораторная работа №3. Обработка деревьев | 2 |
| Лабораторная работа №4. Разработка сетевого сервиса и клиента | 4 |
| Рубежный контроль №2. | 2 |
| Всего часов лабораторных работ | 16 |

5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Конспект лекций (краткий обзор рассматриваемых на лекциях вопросов) представлен в формате мультимедийных презентаций и включен в состав учебно-методического комплекса дисциплины, доступного студентам.

Основная цель выполнения лабораторных работ – получение практических навыков разработки программного обеспечения.

Программой изучения дисциплины предусмотрено выполнение лабораторных работ.

Самостоятельная работа студентов по освоению дисциплины включает подготовку к промежуточной аттестации (зачету), подготовку к проведению рубежного контроля, проработку лекционного материала, выполнение и оформление результатов лабораторных работ.

Для текущего контроля успеваемости обучения используется балльно-рейтинговая система контроля. Для получения высокой оценки настоятельно рекомендуется активно участвовать во время обсуждения материала дисциплины на лекционных и лабораторных занятиях, а также тщательно его прорабатывать при самостоятельной работе.

Рекомендуемая трудоемкость самостоятельной работы

| Виды самостоятельной работы | Рекомендуемая трудоемкость, акад. часов |
|---|---|
| Изучение материала лекционного курса: | 27 |
| Подготовка и выполнение лабораторных работ | 50 |
| Подготовка к рубежному контролю (по 2 часа на контроль) | 4 |
| Подготовка к зачету | 18 |
| Всего: | 108 |

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ
2. Задания для пробного и контрольного тестирования по разделам.
3. Вопросы для подготовки к зачету по дисциплине.
4. Образцы отчетов по практическим заданиям.

6.2 Система балльно-рейтинговой оценки работы студентов

Рейтинговые балльные оценки по дисциплине

| Виды контроля/аттестации по дисциплине | Содержание | Максимальная оценка |
|--|----------------------------------|---------------------|
| Текущий контроль | Посещение лекций (26 за занятие) | 36 x 4 = 8 б |
| | Выполнения практических заданий | 42 б |
| Рубежный контроль | Рубежный контроль №1 | 10 б |
| | Рубежный контроль №2 | 10 б |
| Промежуточная аттестация (зачет) | | 30 б |
| Максимальная итоговая оценка, баллов | | 100 б |

Для допуска к зачету обучающийся должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и при этом он должен выполнить и защитить все лабораторные работы и контрольную работу.

Для получения оценки «зачтено» автоматически обучающемуся достаточно набрать 61 баллов по результатам текущего и рубежного контроля в течение семестра.

Обучающемуся преподаватель вправе добавить до 30 дополнительных (бонусных) баллов за активность на учебных занятиях, оригинальность принимаемых решений при выполнении лабораторных работ и индивидуальных контрольных заданий.

В случае если по результатам текущего и рубежного контроля студентом набрано менее 50 баллов, он может набрать недостающее количество баллов, выполнив дополнительные индивидуальные задания до конца зачетной недели семестра.

Состав дополнительных заданий, их количество, формы выполнения и максимальные балльные оценки определяются преподавателем и доводятся до студента в момент выдачи заданий.

Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, также проводится путем выполнения дополнительных индивидуальных заданий.

6.3 Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме фронтального тестирования по теоретической части дисциплины (тематические разделы №1 и №2). Каждый тест содержит 20 вопросов, расчетное время проведения тестирования – 45 минут. Оценивается количество правильных ответов на задания теста. Обучающийся, ответивший правильно менее, чем на 10 заданий теста, считается не прошедшим тестирование и обязан повторно пройти этот тест во время консультации по дисциплине.

Зачет по дисциплине проводится в форме тестирования по всем разделам дисциплины. Тест содержит 30 вопросов, расчетное время проведения тестирования

ния – 45 минут. Оценивается количество правильных ответов и соответственно вычисляется балльная оценка (по 1 баллу за каждый правильный ответ).

Результаты текущего контроля успеваемости и зачета заносятся преподавателем в зачетно-экзаменационную ведомость, которая сдается в организационный отдел института в день проведения зачета, а также выставляются в зачетную книжку студента.

6.4 Примеры оценочных средств для рубежных контролей и зачета

6.4.1 Примеры вопросов для проведения рубежного контроля №1

Дисциплина «Язык C++»

1. Какой из ниже перечисленных операторов, не является циклом в C++?

- а) do while;
- б) for;
- в) while;
- г) repeat until.

2. Сколько параметров можно передать в деструктор?

- а) не более 10;
- б) не более 3;
- в) не более 15;
- г) нельзя передавать параметры в деструктор.

3. Что выдаст код ниже?

```
char s[] = "hello", t[] = "hello";
if(s == t)
    cout << "True";
```

- а) Ничего не выведет, так как идет сравнение указателей.
- б) Нельзя сравнивать массивы на равенство.
- в) True.
- г) Будет ошибка в коде.

Дисциплина «Язык C#»

1. Какая функция корректно сравнивает две подстроки?

- а) String.Match("hi", "hello");
- б) String.Equal("hi", "hello");
- в) String.Compare("hi", "hello");
- г) String.Check("hi", "hello");

2. При каком условии результат следующего выражения будет равен значению true?

```
int a = 1, b = 5;
bool some = false;
```

- а) (b != 5 || a == 3) || (!some && a > 1);

- б) (some && a != 2) || b > 5;
- в) some || a == 4 || b < 3;
- г) (b > 5 && a <= 3) || (!some || a < 1);
- д) (b <= 5 || a == 3) && some.

3. Массив объявлен следующим образом:

```
int [ , , ] a = new int[5, 2, 3];
```

Чему равно значение a.Length?

- а) 10;
- б) 3;
- в) неопределенно;
- г) 30;
- д) 5.

Дисциплина «Язык Julia»

1. Каких типов в языке Julia не существует?

- а) примитивные;
- б) абстрактные;
- в) составные;
- г) классовые.

2. Имеется код:

```
struct Point{T<:Real}
```

```
    x::T
```

```
    y::T
```

```
end
```

```
p = Point(1,2.5)
```

Будет ли сгенерировано исключение?

- а) да;
- б) нет.

3. Функция intersect ответственна за

- а) объединение множеств;
- б) пересечение множеств;
- в) разность двух множеств.

Дисциплина «Язык Python»

1. Выберите правильную конструкцию

- а) for i in list::;
- б) from i in list:
- в) for i in list
- г) i in for list:.

2. Что будет напечатано при выполнении следующего фрагмента:

```
for j in 'Hi! I\'m mister Robert':
```

```
    if j == '\':
```

```

        print("Найдено")
        break
else:
    print ("Готово")

```

Дисциплина «Язык Kotlin»

1. В чем отличие между var и val?

- а) отличий между ними нет;
- б) val - позволяет установить и изменять значение, var - позволяет установить значение без изменения в дальнейшем;
- в) var - позволяет установить и изменять значение, val - позволяет установить значение без изменения в дальнейшем;
- г) val служит для создания массивов, var для создания переменных.

2. Какого типа данных не существует в Kotlin?

- а) Int.
- б) Все перечисленные типы данных существуют.
- в) Array.
- г) List.
- д) Object.

Дисциплина «Язык Haskell»

1. К базовым типам данных в Haskell НЕ относятся следующие типы:

- а) Integer;
- б) Int;
- в) Double;
- г) Float;
- д) Char;
- е) Bool;
- ж) String;
- з) все это базовые типы данных.

Дисциплина «Язык Java»

1. Что подходит и для интерфейса, и для абстрактного класса? Выбрать все возможные варианты.

- а) Методы должны быть абстрактными
- б) Может содержать public static final переменные
- в) Может использовать ключевое слово extends
- г) Может содержать default методы
- д) Может содержать static методы
- е) Нельзя создать инстанс
- ж) Наследуется от класса Object

2. Какой импорт нужно добавить, чтобы следующий код скомпилировался?

```

public class Imports {
    public void read(Files files) {
        Paths.get("name");
    }
}

```

```

}
}
a) import java.nio.files.*;
б) import java.nio.files.Files;
в) import java.nio.files.Paths;

```

Дисциплина «Язык JavaScript»

1. $x = 1$

$y = '1'$

Что вернет оператор $x == y$

а) true

б) false

2. $x = 1$

$y = '1'$

Что вернет оператор $x === y$

а) true

б) false

Дисциплина «Язык Prolog»

1. Укажите неверное утверждение (один ответ)

а) Если действие выполнено до вызова рекурсии, то это использование "подъема".

б) Рекурсия в Прологе - это вызов предикатом самого себя.

в) Если действие выполнено после вызова рекурсии, то это использование "подъема".

г) В Прологе откат при неуспешном поиске называется бектрекингом.

д) В Прологе рекурсия заменяет оператор цикла.

2. Какое из правил определяет количество элементов в списке? (один ответ)

а) $wr(X) :- X=[] ; X=[H|Y], write(H), nl, wr(Y).$

б) $wr(X,R,Z) :- X=[], Z=R ; X=[H|Y], R1=[H|R], wr(Y,R1,Z).$

в) $wr(X,K) :- X=[], write(K) ; X=[H|Y], K1=K+1, wr(Y,K1).$

г) $wr(X,K,N) :- X=[], N=K ; X=[H|Y], K1=K+H, wr(Y,K1,N).$

д) $wr(["Иванов", "Петров", "Сидоров", "Павлов", "Федоров"]).$

Дисциплина «Язык Go»

1. Привести пример кода с оператором go.

2. Привести пример кода с каналом.

Дисциплина «Язык Erlang»

1. Какое из правил определяет количество элементов в списке

а) `bitwise(List)`

б) `length(List)`

в) `size(List)`

г) `len(List)`

д) `<<List:Size>>`

е) `list:foldl(func(A,Acc)-> Acc++size(A),end,[],[1,2,3])`

2. Какое из утверждений не порождает процесс

- а) `spawn(Module,Func,Arg)`
- б) `apply (Module,Func,Arg)`
- в) `spawn(fun() -> ... end)`
- г) `Module:start_link(...)`
- д) `rpc:call(Node,Module,Func,Arg)`

Дисциплина «Язык Ocaml»

- 1. Привести пример кода с оператором `let`.
- 2. Привести пример кода с объявлением и использованием функции.

6.4.2 Примеры вопросов для проведения рубежного контроля №2

Дисциплина «Язык C++»

1. Какую библиотеку `mlpack` использует для работы с матрицами?

- а) `numpy`;
- б) `armadillo`;
- в) `std::vector`.

2. Какие языки поддерживает `mlpack`?

- а) `c++`;
- б) `python`;
- в) `julia`;
- г) `c#`;
- д) `go`.

3. Какое «направление» машинного обучения не представлено в `mlpack`?

- а) регрессия;
- б) классификация;
- в) кластерный анализ;
- г) обучение с подкреплением.

Дисциплина «Язык C#»

1. За что ответственны мутации в GraphQL?

- а) за модификацию данных на сервере;
- б) за изменение данных на стороне клиента;
- в) изменение структуры данных;
- г) все варианты верны.

2. Интерфейс GraphQL представляет собой абстрактный тип, ...

- а) который устанавливает список полей, которые должны быть реализованы в похожих типах объектов;
- б) который устанавливает список методов, которые должны быть реализованы в похожих типах объектов;
- в) который устанавливает список полей и методов, которые должны быть реализованы в похожих типах объектов.

3. Самодиагностика в GraphQL отвечает за
- возможность запрашивать детали о текущей схеме;
 - тестирование запросов;
 - тестирование связей между типами.

Дисциплина «Язык Julia»

- Flux предназначен для ...
 - построения сложных диаграмм;
 - создания графического пользовательского интерфейса;
 - решения задач машинного обучения;
 - создания экспертных систем продукционного вида.
- Существует ли поддержка искусственных нейронных сетей в Flux.jl?
 - да
 - нет
- Какой пакет содержит методы для вычисления основных характеристик случайных величин на основе выборок?
 - Statistics.
 - MLBase.
 - Melin.
 - Strada.

Дисциплина «Язык Python»

- Подсчитать произведение ненулевых элементов на диагонали прямоугольной матрицы. Для $X = np.array([[1, 0, 1], [2, 0, 2], [3, 0, 3], [4, 4, 4]])$ (Ответ 3)
- К текстовым записям на графике в Matplotlib относятся (возможно несколько ответов):
 - наименования осей;
 - наименование самого графика;
 - текстовые примечания на поле с графиком;
 - легенда;
 - все варианты ответов;

Дисциплина «Язык Kotlin»

- По умолчанию, классы в Kotlin являются:
 - Публичными.
 - Абстрактными.
 - Финализированными.
 - Изолированными.
- Что характерно для чистой функции (возможно несколько ответов):
 - чистая функция всегда возвращает значение;
 - она не выбрасывает исключений;
 - не изменяет данных за пределами своей области видимости;
 - не изменяет свои аргументы;

- д) всегда возвращает одинаковое значение для одних и тех же аргументов;
- е) все варианты ответов;

Дисциплина «Язык Haskell»

1. Укажите стандартные операции из модуля Prelude, которые используются для обработки списков (возможно несколько ответов):

- а) !!
- б) ;;
- в) :
- г) ++
- д) Все варианты ответов.

2. Что понимается под кортежем в Haskell:

- а) то же, что и в Python (отличий между ними нет);
- б) набор значений всегда одного типа (заклЮчЁнных в круглые скобки и разделённых запятыми), который составляет единое самостоятельное значение, участвующее в вычислительном процессе;
- в) набор значений может быть разного типа (заклЮчЁнных в круглые скобки и разделённых запятыми), который составляет единое самостоятельное значение, участвующее в вычислительном процессе;
- г) нет правильного ответа.

Дисциплина «Язык Java»

1. Какой будет результат выполнения кода?

```

1: public class _C {
2: private static int $;
3: public static void main(String[] main) {
4: String a_b;
5: System.out.print($);
6: System.out.print(a_b);
7: } }

```

- а) Compiler error on line 1.
- б) Compiler error on line 2.
- в) Compiler error on line 4.
- г) Compiler error on line 5.
- д) Compiler error on line 6.
- е) *0null*

2. Что произойдет, когда запустим код?

```

Boolean b = null;
if (b) {
    System.out.println("...");
}

```

- а) Код не скомпилируется
- б) *java.lang.NullPointerException* на строке *if (b)*

Дисциплина «Язык JavaScript»

1. Что в консоль выведет программа:

```
let a = { a: 1 }
let b = { a: 1 }
let c = a
console.log(a === b)
console.log(a === c)
```

- а) false, true
- б) true, true
- в) false, false
- г) true, false

2. Какое значение будет иметь x после завершения работы программы:

```
let x = 5
x = (x++, x = addFive(x), x *= 2, x -= 5, x += 10)
function addFive(num) { return num + 5 }
```

- а) 20;
- б) 27;
- в) 37;
- г) 17.

Дисциплина «Язык Prolog»

1. Какой из методов представления знаний используется в Прологе? (один ответ)

- а) продукционные правила;
- б) семантические сети;
- в) фреймы;
- г) логические системы;
- д) нейронные сети.

2. Какой из терминов обозначает часть фрейма для хранения данных? (один ответ)

- а) слот;
- б) прототип;
- в) демон;
- г) предикат;
- д) нотация.

Дисциплина «Язык Go»

1. Что такое горутина?

- а) нить;
- б) легковесная нить;
- в) процесс;
- г) подпрограмма.

2. Что такое канал в Golang ?

- а) способ передачи данных между горутинами с блокировкой

- б) способ передачи данных без блокировкой
- в) способ передачи данных с блокировкой предающей горутины или без таковой.

Дисциплина «Язык Erlang»

1. Что такое майлбокс?
 - а) способ обмена данными;
 - б) актер;
 - в) примитив в контексте легковесной нити, принимающий и обрабатывающий сообщения;
 - г) способ взаимодействия узлов кластера Erlang.

2. Что такое node?
 - а) компьютер сети;
 - б) объект администрирования SysOpа;
 - в) экземпляр виртуальной машины, способный к взаимодействию по сети;
 - г) это такой NODE.JS но для Erlang.

3. Что такое кортеж?
 - а) структура данных;
 - б) структура данных из нескольких компонент;
 - в) структура данных из нескольких компонент без именованного доступа к ее компонентам;
 - г) структура данных из нескольких компонент с именованным доступом к ее компонентам.

Дисциплина «Язык Ocaml»

- 1/ Для чего нужен оператор let?
 - а) объявления константы;
 - б) объявления переменной;
 - в) инициализации процесса.

2. Что такое список?
 - а) структура данных;
 - б) структура данных, имеющая голову и хвост.

6.4.3 Примеры вопросов для подготовки к зачету

Дисциплина «Язык C++»

1. Какие утверждения верны?
 - а) Виртуальные методы могут быть online.
 - б) Деструкторы могут быть виртуальными.
 - в) Виртуальные методы могут быть абстрактными.
 - г) Конструкторы могут быть виртуальными.

2. Что делает данная функция?

```
char * func(char *s){
```

```

    while(*s != `6`) ++s;
    return s;
}

```

- а) подсчитывает количество символов, не равных 6 в строке s;
- б) возвращает указатель на первый встреченный символ 6;
- в) возвращает указатель на следующий символ после символа 6.

Дисциплина «Язык C#»

1. Полиморфная переменная — переменная, которая ...

- а) используется в связке со значимыми типами;
- б) обязательно используется с ключевым словом `static`;
- в) в C# нет полиморфных переменных;
- г) может быть изменена один раз за все время существования;
- д) может принимать значения разных типов.

2. Для организации изменения полей посредством свойств, необходимо использовать неявный параметр:

- а) `return`;
- б) `set`;
- в) `void`;
- г) `value`;
- д) `virtual`.

Дисциплина «Язык Julia»

1. Результатом данного выражения

```
[1:5;] .|> [x->x^2, inv, x->2*x, -, isodd]
```

является ..

- а) возникновение синтаксической ошибки;
- б) вектор с элементами 1, 0.5, 6, -4;
- в) кортеж с элементами 2, 1, 3, -2;
- г) пустое множество.

2. Что вернет выражение ниже?

```
1 < 2 ? v("yes") : v("no")
```

- а) `yes`;
- б) `no`;
- в) `true`;
- г) `false`.

Дисциплина «Язык Python»

1. Подсчитать произведение ненулевых элементов на диагонали прямоугольной матрицы. Для `X = np.array([[1, 0, 1], [2, 0, 2], [3, 0, 3], [4, 4, 4]])` (Ответ 3)

2. К текстовым записям на графике в Matplotlib относятся (возможно несколько ответов):

- а) наименования осей;
- б) наименование самого графика;
- в) текстовые примечания на поле с графиком;

- г) легенда;
- д) все варианты ответов;

Дисциплина «Язык Kotlin»

1. Что такое to в приведенном ниже примере:
 - а) Инфиксная функция, создающая пару (33, 42)
 - б) Ключевое слово Kotlin для создания пары (33, 42)
 - в) Ключевое слово для создания диапазона от 33 до 42
 - г) Опечатка

2. Укажите методы, которые определены для всех классов в Kotlin (возможно несколько ответов):
 - а) equals();
 - б) ToInt();
 - в) hashCode();
 - г) toString();
 - д) NullString();
 - е) IsNull().

Дисциплина «Язык Haskell»

1. Что называется охраняющим выражением в Haskell?
 - а) выражение, накладывающее ограничения на аргументы функции и возвращающее значение True или False;
 - б) такого понятия в Haskell нет;
 - в) выражение, накладывающее ограничения на значение функции и возвращающее значение True или False;
 - г) выражение, накладывающее ограничения на аргументы и значение функции и возвращающее значение True или False.

2. Что называется побочным эффектом функции в Haskell (возможно несколько ответов):
 - а) чтение и модификация значений параметров, переданных в функцию;
 - б) осуществление операций ввода/вывода;
 - в) реагирование на исключительные операции и вызов их обработчиков;
 - г) все перечисленное.

Дисциплина «Язык Java»

1. Как соотносятся методы equals и hashCode объектов java (Контракт equals и hashCode)
 - а) Если объекты равны, то их хэшкоды тоже должны быть равны
 - б) Если хэшкоды равны, то объекты не обязательно равны.
 - в) Все перечисленное

2. Какие из перечисленных переменных являются ссылочными?


```
int[] ints;
long[] longs;
```

- String[] strings;
 Object[] objects;
 а) В 3 и 4 строках
 б) В 1 и 2 строках
 в) Все перечисленное

Дисциплина «Язык JavaScript»

1. Какие методы содержит объект Array, которые являются функциями функционального программирования в javascript (выбрать все)

- а) count
 б) sort
 в) map
 г) filter
 д) reduce

2. Что выведет в консоль функция функционального программирования reduce

```
const strs = ['I', ' ', 'am', ' ', 'Iron', ' ', 'Man']
const result = strs.reduce((acc, currentStr) => acc + currentStr, "")
console.log(result)
```

- а) 'I'
 б) 'Man'
 в) 'I am Iron Man'

Дисциплина «Язык Prolog»

1. Убедитесь в справедливости высказывания «Друг моего друга - мой друг», переведя его с естественного языка на язык логики предикатов.

2. Докажите, что правило modus ponens для исчисления высказываний обоснованно.

3. Покажите, что правило modus tollens $((P \rightarrow Q) \wedge \neg Q) \rightarrow \neg P$ логично.

4. Реализуйте в Visual Prolog решение следующей задачи: «В соревнованиях три первых места заняли А, В и С. какое место занял каждый из них, если В занял не второе и не третье место, а С - не третье?»

5. База данных содержит факты вида данные о студентах. Создать проект, позволяющий вывести список студентов 1-го курса указанной специальности, обучающихся на бюджетной основе/

6. База данных содержит факты вида: student(имя, курс). Создать проект, позволяющий сформировать список студентов 1-го курса/

7. Сформировать список из N элементов, начиная с 2. Каждый следующий на 4 больше предыдущего.

Дисциплина «Язык Go»

1. Расскажите об отличиях горутины и процесса операционной системы

2. Расскажите о буферизованных и небуферизованных каналах и их отличиях

Дисциплина «Язык Erlang»

1. Приведите примеры использования функций `map`, `foldl`, `[... || ...]`, `[Head | Tail]` при обработке списков
2. Какими способами можно вызвать код на исполнение на узле кластера Erlang?

Дисциплина «Язык OCaml»

1. Приведите примеры использования функций `map` и `reduce` при обработке списков.
2. Расскажите о `toolchain` OCaml.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

7.1 Дисциплина «Язык C++»

7.1.1 Основная литература

1. Страуструп Б. Программирование: принципы и практика использования C++.: Пер. с англ. - М: ООО И.Д. Вильямс, 2011. - 1248 с.
2. Липпан, Стенли Б. Лажойе, Жози, Му, Барбара Э. Язык программирования C++. Базовый Курс, 5-е изд. : Пер. с англ. - М.: ООО И.Д. Вильямс, 2014. - 1120 с.
3. Лафоре Р. Объектно-ориентированно-программирование в C++, 4-е изд.: Пер. с англ. - СПб: Питер, 2007. - 928 с.
4. Официальный сайт `mlpack` <https://www.mlpack.org>

7.1.2 Дополнительная литература

1. Галовиц Я. C++17 STL. Стандартная библиотека шаблонов. - СПб.: Питер, 2018. - 432 с.

7.2 Дисциплина «Язык C#»

7.2.1 Основная литература

1. Прайс М. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов. 3-е изд. - СПб.: Питер, 2018. - 640 с.
2. Троелсен Э., Джипикс Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. - СПб: ООО Диалектика, 2018. - 1328 с.
3. Васильев А. Программирование на C# для начинающих. Основные сведения . - М : Эксмо, 2018. - 592 с.

7.2.2 Дополнительная литература

1. Скит Дж. С# для профессионалов: тонкости программирования, 3-е изд. : Пер. с англ. - М.: ООО И.Д. Вильямс, 2014. - 608 с.
2. Документация по С# от Microsoft <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

7.3 Дисциплина «Язык Julia»

7.3.1 Основная литература

1. Шеррингтон М. Осваиваем язык Julia.: Пер. с англ. - М.: ДКМ Пресс, 2017. - 416 с.
2. Официальная документация Julia <https://docs.julialang.org/en/v1>

7.4 Дисциплина «Язык Python»

7.4.1 Основная литература

1. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений: Пособие / Прохоренок Н.А. - СПб:БХВ-Петербург, 2016. - 833 с. ISBN 978-5-9775-3648-6. - Текст : электронный. - URL: <https://new.znanium.com/catalog/product/944629>
2. Прохоренок, Н. А. Python 3. Самое необходимое: Пособие / Прохоренок Н.А., Дронов В.А. - СПб:БХВ-Петербург, 2016. - 464 с. ISBN 978-5-9775-3631-8. - Текст : электронный. - URL: <https://new.znanium.com/catalog/product/944129>
3. Официальный сайт Matplotlib: <https://matplotlib.org/>
4. Официальный сайт NumPy: <https://numpy.org/>

7.4.2 Дополнительная литература

1. Медведев А.А. Изучение языка программирования Python [Электронный ресурс]: методические рекомендации для подготовки бакалавров и специалистов направлений 09.03.03, 09.03.04, 10.05.01, 10.05.03 «Прикладная информатика», «Программная инженерия», «Компьютерная безопасность», «Информационная безопасность» / Министерство науки и высшего образования Российской Федерации, Курганский государственный университет, Кафедра программного обеспечения автоматизированных систем; [сост.: А.А. Медведев]. - Электрон. текстовые дан. (тип файла: pdf; размер: 488 Кб). - Курган: Издательство Курганского государственного университета, 2019. - 43, [1] с.: рис. - Библиогр.: с. 43.
2. Основы языка Python. Сайт кафедры ПОАС: <http://it.kgsu.ru/Python/oglav.html>
3. Библиотека PyQt5. Сайт кафедры ПОАС: http://it.kgsu.ru/Python_Qt/oglav.html

7.5 Дисциплина «Kotlin»

7.5.1 Основная литература

1. Сомон, П. Волшебство Kotlin : практическое руководство / П. Сомон ; пер. с англ. А. Н. Киселева. - Москва : ДМК Пресс, 2020. - 536 с. - ISBN 978-5-97060-801-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1094968>
2. Официальный сайт Kotlin: <https://kotlinlang.org/>

7.5.2 Дополнительная литература

1. Основы языка Kotlin. Сайт кафедры ПОАС: <http://it.kgsu.ru/Kotlin/oglav.html>

7.6 Дисциплина «Haskell»

7.6.1 Основная литература

1. Душкин, Р. В. Функциональное программирование на языке Haskell [Электронный ресурс] / Р. В. Душкин. - Москва : ДМК Пресс, 2008.- 609 с.: ил. - ISBN 5-94074-335-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/409306>
2. Официальный сайт Haskell: <https://www.haskell.org/>

7.6.2 Дополнительная литература

1. Основы языка Haskell. Сайт кафедры ПОАС: <http://it.kgsu.ru/Haskell/oglav.html>

7.7 Дисциплина «Язык Java»

7.7.1 Основная литература

1. Раджпут Д. Spring. Все паттерны проектирования. : Пер. с англ. - Спб.: Питер, 2019. - 320 с.
2. Уоллс К. Spring в действии. : Пер. с англ. - М.: ДМК Пресс, 2015. - 752 с.
3. Козмина Ю., Харроп Р., Шефер К., Хо К. Spring 5 для профессионалов. : Пер. с англ. - Спб.: Диалектика, 2020. - 1119 с.

7.7.2 Дополнительная литература

1. Урма Р., Фуско М., Майкрофт А. Современный язык Java. Лямбда-выражения, потоки и функциональное программирование. : Пер. с англ. - Спб.: Питер, 2020. - 592 с.
2. Шилдт Г. Java. Полное руководство. : Пер. с англ. - Спб.: Диалектика, 2019. - 1488 с.
3. Гутьеррес Ф. Spring Boot 2: лучшие практики для профессионалов. : Пер. с англ. - Спб.: Диалектика, 2020. - 464 с.
4. Официальный сайт spring <https://spring.io/>
5. Официальный сайт создания приложений spring <https://start.spring.io/>

7.8 Дисциплина «Язык JavaScript»

7.8.1 Основная литература

1. Васильев А. JavaScript в примерах и задачах. - М: Эксмо, 2019. - 720 с.
2. Браун Э. Изучаем JavaScript : руководство по созданию современных веб-сайтов. : Пер. с англ. - Спб.: Альфа-книга, 2017. - 368 с.
3. Резиг Д., Фергюсон Р., Пакстон Д. JavaScript для профессионалов. : Пер. с англ. - М.: Вильямс, 2019. - 235 с.

7.8.2 Дополнительная литература

1. Фрисби М. Angular. Сборник рецептов. : Пер. с англ. - Спб.: Альфа-книга, 2018. - 452 с.
2. Дейли Б., Дейли К. Разработка веб-приложений с помощью Node.js, MongoDB и Angular. : Пер. с англ. - М.: Вильямс, 2020. - 656 с.
3. Тиленс М. React в действии. : Пер. с англ. - Спб.: Альфа-книга, 2019. - 368 с.
4. Хэнчетт Э., Листуон Б. Vue.js в действии. : Пер. с англ. - Спб.: Питер, 2020. - 301 с.
5. Заметти Ф. Flutter на практике: Прокачиваем навыки мобильной разработки с помощью открытого фреймворка от Google. : Пер. с англ. - М: ДКМ Пресс, 2020. - 328 с.
6. Официальный сайт Flutter <https://flutter.dev/>

7.9 Дисциплина «Язык Prolog»

7.9.1 Основная литература

1. Цуканова Н. И. Теория и практика логического программирования на языке Visual Prolog 7: Учебное пособие для вузов / Цуканова Н.И., Дмитриева Т.А. - Москва: Гор. линия-Телеком, 2013. - 232 с. (Учебное пособие для высших учебных заведений) ISBN 978-5-9912-0194-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/414543> (дата обращения: 05.06.2020).
2. Основы программирования на языке Пролог. - URL: <https://www.intuit.ru/studies/courses/44/44/info>
3. https://wikiru.visual-prolog.com/index.php?title=%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B_%D0%9F%D1%80%D0%BE%D0%BB%D0%BE%D0%B3%D0%B0.%D0%A7%D0%B0%D1%81%D1%82%D1%8C_1
4. Волчёнков, Н. Г. Логическое программирование. Язык Пролог: учебное пособие / Н. Г. Волчёнков. - 2-е. - Москва : НИЯУ МИФИ, 2015. - 160 с. ISBN 978-5-7262-2091-8. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/126655> (дата обращения: 05.07.2020). - Режим доступа: для авториз. пользователей

7.9.2 Дополнительная литература

1. Могилев А.В. Методы программирования. Компьютерные вычисления: учебное пособие / А. В. Могилев, Л. В. Листрова. - Санкт-Петербург: БХВ-Петербург, 2008. - 320 с. - ISBN 978-5-9775-0151-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/350418> (дата обращения: 05.06.2020).
2. Шрайнер, П. А. Основы программирования на языке Пролог: учебное пособие / П. А. Шрайнер. - 2-е изд. - Москва: ИНТУИТ, 2016. - 213 с. - ISBN 5-9556-0034-5. — Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/100322> (дата обращения: 05.07.2020). - Режим доступа: для авториз. Пользователей.
3. Кокин А.Г. Турбо Пролог [Электронный ресурс]: учебное пособие / А.Г. Кокин; Министерство образования и науки Российской Федерации, Федеральное агентство по образованию, Курганский государственный университет; [науч.ред. В.А. Симахин]. - Электрон. текстовые дан. (тип файла: pdf; размер: 465 Kb). - Курган: Издательство Курганского государственного университета, <http://dspace.kgsu.ru/xmlui/handle/123456789/48932006>. - 93, [1] с.: ил., табл. - Библиогр.: с. 93. - ISBN 5-86328-744-6.

7.10 Дисциплина «Язык Go»

7.10.1 Основная литература

1. <http://www.golangbootcamp.com/> Go Bootcamp Everything you need to know to get started with Go. Matt Aimonetti
2. <https://www.manning.com/books/go-in-action>, Go in Action ,
3. William Kennedy with Brian Ketelsen and Erik St. Martin *Foreword by Steve Francia* ,November 2015 ,ISBN 9781617291784 ,264 pages,Manning

7.10.2 Дополнительная литература

1. <https://proglib.io/p/30-golang-books/> Библиотека программиста, 27 октября 2017 ,30 лучших книг для освоения языка программирования Go
2. The Go Programming Language,Alan A. A. Donovan · Brian W. Kernighan Published Oct 26, 2015 in paperback and Nov 20 in e-book,Addison-Wesley; 380pp; ISBN: 978-0134190440,

7.11 Дисциплина «Язык Erlang»

7.11.1 Основная литература

1. <https://www.kv.by/post/1061811-12-knig-dlya-izucheniya-yazyka-programmirovaniya-erlang> 12 книг для изучения языка программирования Erlang
2. <https://www.erlang.org/> Erlang OTP site
3. <https://www.erlang.org/docs> Erlang OTP Documentation
4. Программирование в Erlang, Франческо Чезарини ,Серия: Функциональное программирование ,2012 , ДМК-Пресс

7.11.2 Дополнительная литература

1. <https://github.com/search?q=Erlang> проекты Erlang на github
2. <https://habr.com/ru/post/50028/> Erlang. Что это, зачем, как и для кого.

7.12 Дисциплина «Язык OCaml»

7.12.1 Основная литература

1. <https://ocaml.org/learn/books.html> Ocaml books
2. <https://sbermegamarket.ru/catalog/details/programmirovanie-na-yazyke-ocaml-100023310459/> Программирование на языке OCaml

7.12.2 Дополнительная литература

1. <https://github.com/search?q=OCaml> проекты Ocaml на github
2. <https://ocaml.org/learn/teaching-ocaml.html> Teaching OCaml

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Техническое обеспечение

| № | Наименование | Использование |
|---|---|---|
| 1 | Комплект: ноутбук, медиа-проектор, экран | Для демонстрации иллюстративного материала при чтении лекций. |
| 2 | Персональный компьютер стандартной комплектации | Используется в качестве инструмента и объекта исследования при выполнении лабораторных и контрольных работ. |

8.2 Программное обеспечение

| № | Наименование | Использование |
|---------------|--|--|
| C++ | | |
| 1 | Компилятор gcc | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | Среда разработки CLion | |
| 3 | Библиотека matplotlib-cpp | |
| 4 | Библиотека mplack | |
| C# | | |
| 1 | Среда разработки Microsoft Visual Studio Community | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | Библиотека GraphQL for .NET | Программные средства для выполнения лабораторной работы №2 |
| 3 | SQLite | |
| Julia | | |
| 1 | Juno | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | JuliaStats | Программные средства для выполнения лабораторной работы №2 |
| 3 | Flux | |
| Python | | |
| 1 | Python 3.6 или выше, | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | JetBrains PyCharm Community Edition 2018.3.7 или выше, | |
| 3 | PyQt5 | |

| | | |
|-------------------|--|---|
| 4 | matplotlib: 3.0.2 или выше | |
| 5 | NumPy: 1.15.4 или выше | |
| 6 | VisPy: 0.5 или выше | |
| 7 | MayaVi: 4.7 или выше | |
| Kotlin | | |
| 1 | kotlin 1.4 или выше. | Среда IntelliJ IDEA для разработки приложений (все лабораторные работы) |
| Haskell | | |
| 1 | Haskell IDE Engine | Среда IntelliJ IDEA для разработки приложений (все лабораторные работы) |
| Java | | |
| 1 | JDK версии 1.8 и выше. | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | IDE IntelliJ IDEA | Программные средства для выполнения лабораторной работы №2 |
| 3 | Spring Boot 2.0 | |
| 4 | Database H2 | |
| JavaScript | | |
| 1 | WebStorm JetBrains или IDE IntelliJ IDEA | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | React | Программные средства для выполнения лабораторной работы №2 |
| 3 | Vue | |
| 4 | Angular | |
| 5 | Google flutter | |
| Go | | |
| 1 | Компилятор Golang | Выполнение лабораторных работ |
| Erlang | | |
| 1 | Erlang OTP | Выполнение лабораторных работ |
| OCaml | | |
| 1 | Компилятор OCaml | Выполнение лабораторных работ |
| Prolog | | |
| 1 | SWI-Prolog | Программные средства для выполнения лабораторных работ №1 и №2 |
| 2 | Visual-Prolog | |

Аннотация
рабочей программы модуля

ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

образовательной программы высшего образования –
программы магистратуры

09.04.04 Программная инженерия
направленность

*Методы и алгоритмы интеллектуальной обработки данных
в информационно-вычислительных системах*

формы обучения – очная

Трудоемкость освоения модуля – 12 зач. ед. (432 акад. часов)

Семестры: 1-й и 2-й

Содержание модуля

Модуль «Языки и технологии программирования» содержит 12 элективных дисциплин, 6 из которых (по три дисциплины в первом и втором семестрах) должны быть выбраны студентами для изучения:

| | |
|---------------------|-------------|
| <u>1-й семестр:</u> | Язык Julia |
| Язык Python | Язык C# |
| Язык C++ | Язык Kotlin |
| Язык Java | Язык Go |
| Язык JavaScript | Язык Prolog |
| Язык Haskell | Язык Ocaml |
| Язык Erlang | |

2-й семестр:

Язык C++

Раздел 1. Основы языка C++

Раздел 2. Библиотеки для решения задач машинного обучения

Язык C#

Раздел 1. Основы языка C#

Раздел 2. GraphQL

Язык JULIA

Раздел 1. Основы языка Julia

Раздел 2. Библиотеки для решения задач машинного обучения

Язык PYTHON

Раздел 1. Особенности языка и реализации Python-приложений

Раздел 2. Обзор Python-библиотек для работы с большими массивами данных

Язык KOTLIN

Раздел 1. Основы Kotlin

Раздел 2. Объектно-ориентированная и функциональная парадигмы в Kotlin

ЯЗЫК HASKELL

Раздел 1. Основы Haskell

Раздел 2. Списки в Haskell

Раздел 3. Конкурентное и параллельное программирование в Haskell

ЯЗЫК JAVA

Раздел 1. Основы языка Java

Раздел 2. Java фреймворк Spring

ЯЗЫК JAVASCRIPT

Раздел 1. Основы языка Javascript

Раздел 2. Javascript фреймворки

ЯЗЫК PROLOG

Раздел 1. Основы языка Prolog

Раздел 2. Основы теории экспертных систем

ЯЗЫК ERLANG

Раздел 1. Основы Erlang. Парадигмы и синтаксис.

Раздел 2. Основы распределенных систем на Erlang OTP

ЯЗЫК GO

Раздел 1. Язык Golang , его парадигма и синтаксис

Раздел 2. Взаимодействие процессов в Golang

ЯЗЫК OCAML

Раздел 1. OCaml: пардигма и синтаксис

Раздел 2. Разработка кода на OCaml