

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
"Казанский (Приволжский) федеральный университет"
Высшая школа информационных технологий и интеллектуальных систем



подписано электронно-цифровой подписью

Программа дисциплины
Разработка API для мобильных приложений Б1.В.ДВ.10

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: не предусмотрено

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Бажанов В.А.

Рецензент(ы):

Хайруллин А.Ф.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Хасьянов А. Ф.

Протокол заседания кафедры No ____ от " ____ " _____ 201__ г

Учебно-методическая комиссия Высшей школы информационных технологий и интеллектуальных систем:

Протокол заседания УМК No ____ от " ____ " _____ 201__ г

Регистрационный No 689515919

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) Бажанов В.А. , VABazhanov@kpfu.ru

1. Цели освоения дисциплины

Целями освоение данной дисциплины являются:

- Изучить язык программирования Ruby на уровне уверенного владения,
- Изучить основные методы и инструменты создания серверных приложений на Ruby on Rails,
- Изучить современные методы построения программных интерфейсов т.н. API
- Изучить современные методы реализации интерфейсов API:
- RESTful API
- JSON API
- GraphQL
- Изучить методы тестирования и поддержки качества программного кода

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.В.ДВ.10 Дисциплины (модули)" основной образовательной программы 09.03.03 Прикладная информатика и относится к дисциплинам по выбору. Осваивается на 4 курсе, 7 семестр.

Для изучения дисциплины студент должен предварительно успешно изучить следующие курсы:

- Информатика
- Вычислительные системы, сети и телекоммуникации
- Операционные системы
- Мобильные информационные системы

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-11 (профессиональные компетенции)	Способность эксплуатировать и сопровождать информационные системы и сервисы
ПК-15 (профессиональные компетенции)	Способность осуществлять тестирование компонентов информационных систем по заданным сценариям
ПК-17 (профессиональные компетенции)	Способность принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла
ПК-18 (профессиональные компетенции)	Способность принимать участие в организации ИТ-инфраструктуры и управлении информационной безопасностью

В результате освоения дисциплины студент:

1. должен знать:
 - Базовые принципы функционирования сети Интернет

- Язык программирования Ruby на уровне, достаточном для самостоятельной разработки ПО на нём
- Основные идеи методологии командной разработки ПО 'Agile'
- Устройство и предназначение фреймворка Ruby on Rails
- Способы использования фреймворка Ruby on Rails для быстрого прототипирования приложений
- Использование системы контроля версий GIT
- Использование системы управления базами данных Postgresql

2. должен уметь:

- Создать БД и настроить параметры доступа к ней, используя СУБД Postgresql
- Правильно установить и настроить интерпретатор языка Ruby
- Создать репозиторий GIT и использовать его для ведения истории разработки
- Создать базовое web-приложение с помощью инструментов фреймворка Ruby on Rails
- Добавлять новую функциональность в приложение с помощью генераторов фреймворка Ruby on Rails
- Создавать контроллеры, взаимодействующие с мобильными приложениями посредством формата JSON
- Тестировать программный код для обеспечения надёжности кода

3. должен владеть:

- Терминологией, связанной с разработкой ПО для Интернет
- ОС семейства UNIX (GNU/Linux, OS X) на уровне 'уверенный пользователь' и выше
- Утилитами командной строки (ls, cd, pwd, ln, echo, export, rm и т.п.)
- Инструментами и утилитами фреймворка Ruby on Rails
- Утилитами в составе языка Ruby (irb, rake)
- ПО для управления командной разработкой (Trello, PivotalTracker, Asana)

4. должен демонстрировать способность и готовность:

- Использовать язык программирования Ruby для разработки прототипов ПО
- Использовать методологии командной разработки ПО 'Agile'
- Знать внутреннее устройство фреймворка Ruby on Rails
- Использовать фреймворк Ruby on Rails для быстрого прототипирования приложений
- Использовать систему контроля версий GIT
- Использовать системы управления базами данных Postgresql
- Владеть терминологией, связанной с разработкой ПО для Интернет
- Владеть азами администрирования ОС семейства UNIX (GNU/Linux, OS X)
- Владеть инструментами и утилитами фреймворка Ruby on Rails
- Пользоваться утилитами в составе языка Ruby;
- Использовать ПО для управления командной разработкой.

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 5 зачетных(ые) единиц(ы) 180 часа(ов).

Форма промежуточного контроля дисциплины: зачет с оценкой в 7 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабораторные работы	
1.	Тема 1. Знакомство с протоколом передачи данных HTTP, виды запросов, заголовки и куки	7		0	0	6	
2.	Тема 2. Использование различных типов HTTP запросов для выполнения операций с данными через Интернет	7		0	0	6	
3.	Тема 3. Знакомство с языком программирования Ruby, ООП в Ruby	7		0	0	6	Контрольная работа
4.	Тема 4. Знакомство с архитектурой MVC и особенностями ООП языка Ruby	7		0	0	6	
5.	Тема 5. Знакомство с основами тестирования кода	7		0	0	6	
6.	Тема 6. Основы построения веб-приложений с помощью фреймворка Ruby on Rails	7		0	0	6	Письменное домашнее задание
7.	Тема 7. Основы построения веб-приложений с помощью фреймворка Ruby on Rails(продолжение)	7		0	0	6	
8.	Тема 8. Rails и базы данных, операции с данными через HTTP и операции с данными в БД, аналогии CRUD	7		0	0	6	
9.	Тема 9. Разработка API - основы	7		0	0	6	
10.	Тема 10. Разработка API - особенности разработки для мобильных приложений	7		0	0	6	Письменная работа
11.	Тема 11. JSON API, GraphQL - современные подходы к реализации API	7		0	0	6	Письменное домашнее задание

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
12.	Тема 12. Тестирование API	7		0	0	6	Письменное домашнее задание
.	Тема . Итоговая форма контроля	7		0	0	0	Зачет с оценкой
	Итого			0	0	72	

4.2 Содержание дисциплины

Тема 1. Знакомство с протоколом передачи данных HTTP, виды запросов, заголовки и куки

лабораторная работа (6 часа(ов)):

В рамках темы студенты знакомятся с понятием "сетевой протокол обмена данными", изучают историю создания первых протоколов семейства TCP/IP, изучают состояния протокола HTTP, способы обмена данными между серверов и клиентом в локальной и глобальной компьютерной сети, рассматривают типы запросы от клиента серверу, виды ответов сервера клиенту, изучают такие понятий, как заголовки и cookies. Тема подготавливает студентов к более детальному изучению протокола HTTP как основного способа передачи информации в сети Интернет.

Тема 2. Использование различных типов HTTP запросов для выполнения операций с данными через Интернет

лабораторная работа (6 часа(ов)):

В рамках темы студенты более детально изучают протокол обмена информацией HTTP, способы выполнения различных операций над данными, находящимися на удалённом сервере в сети, изучают существующие стандарты и соглашение, принятые в разработке клиент-серверных приложений, рассматривают основные проблемы стандартизации операций над данными и способы их решения. Тема поясняет принцип работы компьютерных сетей в целом, даёт общую картину функционирования клиент-серверных приложений.

Тема 3. Знакомство с языком программирования Ruby, ООП в Ruby

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают основы языка программирования Ruby. Изучается история и причины его создания, основные особенности, отличия от других популярных языков программирования. Рассматриваются способы установки интерпретатора Ruby в операционной системе Linux / Mac OS. Рассматривается проблема использования нескольких версий интерпретатора для разных проектов. Студенты изучают синтаксис, семантику языка, стандартные типы данных и функции, доступные в стандартной библиотеке. Так же изучаются способы запуска приложений, написанных на Ruby.

Тема 4. Знакомство с архитектурой MVC и особенностями ООП языка Ruby

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают основы т.н. объектно-ориентированного программирования, подходы к построению ПО с применением этого подхода. Изучаются особенности реализации объектно-ориентированного подхода к разработке ПО в Ruby, изучаются такие понятия, как Классы, Объекты, Наследование, Модули, рассматриваются модификаторы доступа к методам класса public, private и protected. Рассматриваются механизмы загрузки интерпретатором файлов с исходным кодом, изучаются ключевые слова require, include и load, способы и случаи их применения.

Тема 5. Знакомство с основами тестирования кода

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают основы тестирования кода, способы написания тестов для кода на языке Ruby, особенности выполнения кода, тестирующего другой код. Изучается вопрос повышения надёжности кода в условиях командной разработки с помощью написания и регулярного запуска тестирующего кода, рассматривается роль тестов в качестве дополнительного описания основного кода. Студенты знакомятся с инструментом тестирования: RSpec, пишут тесты для кода своих проектов, изучают такие понятия, как Stub и Mock.

Тема 6. Основы построения веб-приложений с помощью фреймворка Ruby on Rails

лабораторная работа (6 часа(ов)):

В рамках темы студенты знакомятся с веб-фреймворком Ruby on Rails, изучают историю и причины его создания, изучают проблемы, которые призван решить фреймворк. Студенты знакомятся со способами быстрого построения веб-приложений инструментами Ruby on Rails, рассматривают входящие в фреймворк скрипты-генераторы: scaffold, resource, controller, model. Изучается структура приложения, использующего Rails, его составные части, способы запуска, остановки, контроля логов и базовая настройка его компонентов.

Тема 7. Основы построения веб-приложений с помощью фреймворка Ruby on Rails(продолжение)

лабораторная работа (6 часа(ов)):

В рамках темы студенты продолжают знакомство с построением веб-приложений инструментами Ruby on Rails. Изучается жизненный цикл HTTP-запроса, его обработка последовательно разными частями фреймворка: использование конфигурации маршрутов, сопоставление типов запросов определённому контроллеру, написание кода "бизнес-логики", формирования ответа в виде HTML-документа, использования шаблонов для создания динамического контента страниц. Изучается принцип "соглашения важнее конфигурации".

Тема 8. Rails и базы данных, операции с данными через HTTP и операции с данными в БД, аналогии CRUD

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают конфигурацию подключения проекта на Rails к СУБД, на примере SQLite 3 и PostgreSQL 10.x. Изучается способ работы с БД, добавления, изменения, поиска и удаления данных. Изучаются разные подходы, реализующие эти действия, изучается понятие Data Mapping и ORM. Студенты рассматривают библиотеку ActiveRecord, её методы для создания, изменения, удаления данных, а также рассматривается операция поиска данных, в том числе "нечёткого поиска". Студенты учатся выполнению SQL запросов через стандартные утилиты СУБД, через методы ActiveRecord, через написание запросов вручную, кратко изучается вопрос об оптимизации времени выполнения запросов. Отдельно рассматривается обработка ошибок при выполнении запроса к БД.

Тема 9. Разработка API - основы

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают проблему создания Application Programming Interface в широком смысле этого понятия, как интерфейса для межпрограммного взаимодействия. Изучается история проблемы, её решения на других платформах, рассматривается для чего используется, где применяется API, основные идеи и правила построения программных интерфейсов. Затем изучается способ построения API для Web-приложений, изучается формат JSON, такие понятия как REST API и его внутреннее устройство.

Тема 10. Разработка API - особенности разработки для мобильных приложений

лабораторная работа (6 часа(ов)):

В рамках темы студенты изучают создание API-приложения на Rails, реализующего сервер данных для мобильного приложения. Изучаются такие особенности такого рода приложения, как версионность протокола API, аутентификация пользователей в помощью токенов, правила реализации интерфейсов для минимизации трафика и одновременной минимизации количества запросов к серверу, необходимых для получения нужной информации. Рассматривается REST API как наиболее распространённый формат протокола, изучается его структура, плюсы и недостатки. Студенты реализуют API контроллеры в своих приложениях, изучают дополнительное ПО для разработки вроде Postman.

Тема 11. JSON API, GraphQL - современные подходы к реализации API лабораторная работа (6 часа(ов)):

В рамках темы изучается детально способы организации API известный как RESTfull API, рассматриваются его преимущества и недостатки, поднимается вопрос об оптимизации количества пересылаемых данных, уменьшения задержек. Изучения плюсов и минусов использования REST API для различных типов приложений с различной структурой данных. Рассматривается попытка стандартизации API известная как JSON API. Изучается менее распространённый способ организации API - GraphQL, его отличия и преимущества.

Тема 12. Тестирование API

лабораторная работа (6 часа(ов)):

В рамках темы изучается вопрос о необходимости автоматизированного тестирования ПО, рассматривается случай тестирования приложения, предоставляющего Application Programming Interface, рассматриваются отличия этого процесса от тестирования простых web-приложений, изучаются способы повышения надёжности приложений с API. Студенты изучают методики тестирования, рассматривают существующее ПО, автоматизирующее создание наборов тестов, применяют его в своих проектах. Анализируется роль тестирования в процессе описания требований к ПО. Студенты изучают методологию Test Driven Development, рассматривают плюсы подхода, предполагающего написание тестов до написания кода.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
3.	Тема 3. Знакомство с языком программирования Ruby, ООП в Ruby	7		подготовка к контрольной работе	20	Контрольная работа
6.	Тема 6. Основы построения веб-приложений с помощью фреймворка Ruby on Rails	7		подготовка домашнего задания	24	Письменное домашнее задание
10.	Тема 10. Разработка API - особенности разработки для мобильных приложений	7		подготовка к письменной работе	16	Письменная работа

№	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
11.	Тема 11. JSON API, GraphQL - современные подходы к реализации API	7		подготовка домашнего задания	24	Письменное домашнее задание
12.	Тема 12. Тестирование API	7		подготовка домашнего задания	24	Письменное домашнее задание
	Итого				108	

5. Образовательные технологии, включая интерактивные формы обучения

На лабораторных работах предполагается разбиение студентов на группы для командной работы.

Демонстрация преподавателем реализации рассматриваемых алгоритмов, существующих технологий и результатов прочих упражнений, выполняемых во время занятий, с помощью мультимедийного проектора.

Парное программирование (один человек набирает код, второй говорит, что писать)

Командное программирование (разделение обязанностей и функций по разработке программ - по модулям или технологиям)

Мастер-классы по определенным темам (повторение практических вещей для преподавателей)

Использование системы контроля версий (технология, используемая во всех компаниях, занимающихся промышленным программированием).

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

Тема 1. Знакомство с протоколом передачи данных HTTP, виды запросов, заголовки и куки

Тема 2. Использование различных типов HTTP запросов для выполнения операций с данными через Интернет

Тема 3. Знакомство с языком программирования Ruby, ООП в Ruby

Контрольная работа, примерные вопросы:

Задание: выполнить самостоятельную работу, разработать ПО, согласно индивидуальному заданию преподавателя. Использовать язык Ruby и объектно-ориентированный подход к проектированию. Выложить результат на Github и прислать ссылку на репозиторий преподавателю. Вопросы: - Опишите основные свойства и особенности языка Ruby, основы синтаксиса и семантики языка. - Опишите, что такое объектно-ориентированный подход в программировании - Опишите реализацию объектно-ориентированного подхода в языке Ruby - Опишите, какие утилиты входят в комплект поставки Ruby, для чего используются. - Опишите возможности стандартной библиотеки Ruby. - Опишите, как вы видите сферу применения языка Ruby. - Продемонстрируйте, каким образом установлен интерпретатор языка Ruby на вашем компьютере. - Опишите проблему необходимости установки нескольких версий интерпретатора и её решение. - Опишите, как работает утилита rbenv, ruby-build. - Опишите, как работает инициализация rbenv в командном интерпретаторе bash.

Тема 4. Знакомство с архитектурой MVC и особенностями ООП языка Ruby

Тема 5. Знакомство с основами тестирования кода

Тема 6. Основы построения веб-приложений с помощью фреймворка Ruby on Rails

Письменное домашнее задание , примерные вопросы:

Задание: выполнить самостоятельную работу, разработать клиент-серверное ПО на основе фреймворка Ruby on Rails, согласно индивидуальному заданию преподавателя. Использовать язык Ruby, фреймворк Ruby on Rails. Выложить результат на Github и прислать ссылку на репозиторий преподавателю. Вопросы: - Что такое фреймворк? Какие компоненты содержит "веб-фреймворк? Перечислите их функции. - Опишите, что такое Ruby on Rails, какие компоненты его вы знаете? - Опишите жизненный цикл HTTP-запроса, от его генерации браузером для передачи браузеру готовой HTML-страницы. - Опишите, что происходит с запросом внутри Rails. - Опишите, для чего нужен шаблонизатор и как он используется в Rails. - Опишите, как Rails работает с СУБД на примере SQLite или PostgreSQL - Опишите паттерн MVC - Опишите, какую роль играют классы "Модели" в Rails? - Опишите, какую роль играют классы "Контроллеры" в Rails? - Опишите кратко жизненный цикл HTTP запроса, его обработки в Rails

Тема 7. Основы построения веб-приложений с помощью фреймворка Ruby on Rails(продолжение)

Тема 8. Rails и базы данных, операции с данными через HTTP и операции с данными в БД, аналогии CRUD

Тема 9. Разработка API - основы

Тема 10. Разработка API - особенности разработки для мобильных приложений

Письменная работа , примерные вопросы:

Задание: выполнить самостоятельную работу, добавить API в ПО на основе фреймворка Ruby on Rails, согласно индивидуальному заданию преподавателя. Использовать язык Ruby, фреймворк Ruby on Rails и REST API. Выложить результат на Github и прислать ссылку на репозиторий преподавателю. Вопросы: - Что такое API? Опишите цель создания и область применения API. - Опишите, что такое клиент-серверная архитектура. - Приведите примеры межпрограммного взаимодействия, где используется создание API для подключения программ-клиентов к программам-серверам? - Какие параметры должен иметь "хороший" сервер API? - Опишите, каким образом передаётся информация через API? Какие вы знаете способы стандартизировать передачу данных через API? - Опишите, как вы понимаете, что такое REST? - Опишите, какие недостатки есть на сегодняшний день у приложений, реализующих RESTful API - Опишите, в чем основная идея JSON API? - Расскажите, какие вы знаете иные способы взаимодействия между процессами и модулями ПО

Тема 11. JSON API, GraphQL - современные подходы к реализации API

Письменное домашнее задание , примерные вопросы:

Задание: выполнить самостоятельную работу, добавить API в ПО на основе фреймворка Ruby on Rails, согласно индивидуальному заданию преподавателя. Использовать язык Ruby, фреймворк Ruby on Rails и GraphQL. Выложить результат на Github и прислать ссылку на репозиторий преподавателю. Вопросы: - Что такое GraphQL? - Опишите, что такое клиент-серверная архитектура. - Приведите примеры межпрограммного взаимодействия, где используется создание API для подключения программ-клиентов к программам-серверам? - Какие параметры должен иметь "хороший" сервер API? - Опишите, каким образом передаётся информация через API? Какие вы знаете способы стандартизировать передачу данных через API? - Опишите, как вы понимаете, что такое REST? - Опишите, в чем основная идея GraphQL - Расскажите, какие вы знаете иные способы взаимодействия между процессами и модулями ПО

Тема 12. Тестирование API

Письменное домашнее задание , примерные вопросы:

Задание: выполнить самостоятельную работу, добавить автоматические тесты в свой проект. Использовать библиотеку RSpec. Выложить результат на Github и прислать ссылку на репозиторий преподавателю. Вопросы: - Зачем нужно тестировать проект? - Как тестирование помогает в разработке проекта, в том числе, в командной? - Опишите, каким образом тесты могут выявлять побочные эффекты кода? - Опишите, как работает библиотека RSpec, как её можно применять для тестирования кода? - Опишите, как вы считаете, какой код нужно тестировать, а какой - не нужно. - Опишите, что такое TDD, в чем его основная идея и как вы её понимаете? - Опишите, какое место занимает тестирование в процессе разработки ПО, в какой момент тесты составляются и кем. - Опишите, как вы понимаете, какой код нужно тестировать, а какой код тестировать не обязательно (в Rails приложении) - Расскажите, как вы понимаете, какой код наиболее пригоден для тестирования, а какой код тестировать затруднительно? - Опишите суть методологии BDD и её преимущества перед другими методиками взаимодействия с клиентом.

Итоговая форма контроля

зачет с оценкой (в 7 семестре)

Примерные вопросы к итоговой форме контроля

1. Что такое протокол передачи данных?
2. Протокол HTTP, виды запросов, заголовки
3. Основы языка Ruby, основные конструкции языка.
4. Объектные возможности языка Ruby.
5. Предназначение систем контроля версий.
6. Что такое web-фреймворк, какие проблемы решает его использование.
7. MVC и его реализация в Rails
8. Взаимодействие СУБД и Rails.
9. Модели в Rails.
10. Контроллеры в Rails.
11. Представления в Rails.
12. Основы создания API на Rails
13. Основы систем контроля версий.
14. Способы интернационализации приложений.
15. Установка приложений Rails на heroku.com.
16. Способы защиты от несанкционированного доступа к данным, применяемые в Rails.
17. Тестирование API приложений с помощью инструментов RSpec и Postman
18. JSON API как спецификация структуры данных
19. GraphQL - язык запросов, какие проблемы он призван решать, какие сложности добавляет в проект.
20. MessagePack и его аналоги и их использование в API приложениях.

7.1. Основная литература:

Гаврилова И.В., Разработка приложений [Электронный ресурс] / Гаврилова И.В. - М. : ФЛИНТА, 2017. - 242 с. - ISBN 978-5-9765-1482-9 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785976514829.html>

Тарасов С.В., СУБД для программиста. Базы данных изнутри [Электронный ресурс] / Тарасов С. В. - М. : СОЛОН-ПРЕСС, 2015. - 320 с. - ISBN 978-2-7466-7383-0 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9782746673830.html>

Синаторов С.В., Информационные технологии [Электронный ресурс] / С.В. Синаторов - М. : ФЛИНТА, 2016. - 448 с. - ISBN 978-5-9765-1717-2 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785976517172.html>

Антониоу Г., Семантический веб [Электронный ресурс] / Антониоу Г., Грос П., Хармелен ван Ф., Хоекстра Р. - М. : ДМК Пресс, 2016. - 240 с. - ISBN 978-5-97060-333-8 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785970603338.html>

Калиногорский Н.А., Основы практического применения интернет-технологий [Электронный ресурс] / Калиногорский Н.А. - М. : ФЛИНТА, 2015. - 182 с. - ISBN 978-5-9765-2302-9 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785976523029.html>

7.2. Дополнительная литература:

Создание Web-сайта. Недостающее руководство: Пособие / Мак-Дональд М., - 3-е изд. - СПб:БХВ-Петербург, 2013. - 624 с. ISBN 978-5-9775-0806-3- Режим доступа: <http://znanium.com/catalog.php?bookinfo=943378>

Макарова Н.В., Информатика [Электронный ресурс] : учебник / Под ред. проф. Н.В. Макаровой. - 3-е перераб. изд. - М. : Финансы и статистика, 2009. - 768 с. - ISBN 978-5-279-02202-0 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785279022020.html>

Пархимович М.Н., Основы интернет-технологий [Электронный ресурс] / Пархимович М.Н. - Архангельск : ИД САФУ, 2014. - 366 с. - ISBN 978-5-261-00827-9 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785261008279.html>

7.3. Интернет-ресурсы:

Документация по Git - <https://git-scm.com/doc>

Документация по PostgreSQL - <http://www.postgresql.org/docs/>

Документация по Rails - <http://guides.rubyonrails.org/>

Документация по Ruby - <http://ruby-doc.org/>

Спецификации JSON API - <https://jsonapi.org>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Разработка API для мобильных приложений" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен студентам. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, УМК, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) нового поколения.

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "Консультант студента", доступ к которой предоставлен студентам. Электронная библиотечная система "Консультант студента" предоставляет полнотекстовый доступ к современной учебной литературе по основным дисциплинам, изучаемым в медицинских вузах (представлены издания как чисто медицинского профиля, так и по естественным, точным и общественным наукам). ЭБС предоставляет вузу наиболее полные комплекты необходимой литературы в соответствии с требованиями государственных образовательных стандартов с соблюдением авторских и смежных прав.

Необходим компьютерный класс с проектором.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 09.03.03 "Прикладная информатика" и профилю подготовки не предусмотрено .

Автор(ы):

Бажанов В.А. _____

"__" _____ 201__ г.

Рецензент(ы):

Хайруллин А.Ф. _____

"__" _____ 201__ г.