

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Цыбиков Бэликто Батоевич
Должность: Ректор
Дата подписания: 30.09.2026 10:05:22
Уникальный программный ключ:
056af948c3e48c6f3c571e429957a8ae7b757ae8

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Бурятская государственная сельскохозяйственная академия имени В.Р.
Филиппова»**

Экономический факультет

«СОГЛАСОВАНО»
Заведующий выпускающей
кафедрой
Информатика и информационные
технологии
к.ф.-м.н., доцент

Садуев Н.Б.

подпись
«23» января 2026

«УТВЕРЖДЕНО»
Декан
Экономический факультет
к.э.н., доцент

Баниева М.А.

подпись
«23» января 2026

**Оценочные материалы
Дисциплины (модуля)
Б1.В.11 Программирование на Ruby
Направление подготовки
09.03.03 Прикладная информатика
Направленность (профиль)
Прикладная информатика в экономике АПК**

Улан – Удэ, 2026

ВВЕДЕНИЕ	
1. Оценочные материалы по дисциплине (модулю) являются обязательным обособленным приложением к Рабочей программе дисциплины (модуля) и представлены в виде оценочных средств.	
2. Оценочные материалы является составной частью нормативно-методического обеспечения системы оценки качества обучения обучающимися указанной дисциплины (модуля).	
3. При помощи оценочных материалов осуществляется контроль и управление процессом формирования обучающимися компетенций, из числа предусмотренных ФГОС ВО в качестве результатов освоения дисциплины (модуля).	
4. Оценочные материалы по дисциплине (модулю) включают в себя:	
- оценочные средства, применяемые при промежуточной аттестации по итогам изучения дисциплины (модуля).	
- оценочные средства, применяемые в рамках индивидуализации выполнения, контроля фиксированных видов ВАРО;	
- оценочные средства, применяемые для текущего контроля;	
Перечень видов оценочных средств	

Перечень экзаменационных вопросов
Контрольные вопросы для проведения устных опросов
Комплект заданий для лабораторных работ
Кейс-задания
Комплект тестовых заданий
Комплект заданий для самостоятельной работы обучающихся

Средства для промежуточной аттестации по итогам изучения дисциплины
Нормативная база проведения промежуточной аттестации обучающихся по результатам изучения дисциплины: Программирование на Ruby
1) действующее «Положение о текущем контроле успеваемости и промежуточной аттестации обучающихся ФГБОУ ВО Бурятская ГСХА»

Основные характеристики промежуточной аттестации обучающихся по итогам изучения дисциплины (модуля)	
1	2
Цель промежуточной аттестации -	установление уровня достижения каждым обучающимся целей обучения по данной
Форма промежуточной аттестации -	Экзамен
Место экзамена в графике учебного процесса:	1) подготовка к экзамену и сдача экзамена осуществляется за счёт учебного времени (трудоемкости), отведённого на экзаменационную сессию для обучающихся, сроки 2) дата, время и место проведения экзамена определяется графиком сдачи экзаменов, утверждаемым деканом факультета (директором института)
Форма экзамена -	устный
Процедура проведения экзамена -	представлена в оценочных материалах по дисциплине
Экзаменационная программа по учебной дисциплине:	1) представлена в оценочных материалах по дисциплине 2) охватывает все разделы дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины (модуля)

1. Контрольные вопросы и задания для текущего контроля

Тема 1. Знакомство с Ruby

Вопросы:

Каковы основные синтаксические особенности языка Ruby?

Какие базовые конструкции языка Ruby вы знаете? Приведите примеры.

Задание:

Напишите простую программу на Ruby, выводящую на экран строку и результат арифметического выражения.

Тема 2. Основы Ruby

Вопросы:

Что такое блоки в Ruby, как они используются?

Как осуществляется передача аргументов в методы?

Какие базовые типы данных есть в Ruby?

Приведите пример динамической типизации в Ruby.

Задание:

Реализуйте функцию, принимающую блок и вызывающую его с разными параметрами.

Тема 3. Контроль версий с GIT

Вопросы:

Для чего используются системы контроля версий?

Какие основные преимущества даёт использование Git?

Опишите процесс создания репозитория и отправки изменений на GitHub.

Задание:

Создайте собственный репозиторий на GitHub, выполните коммит и отправьте ссылку преподавателю.

Тема 4. Классы, ООП

Вопросы:

Что такое класс в Ruby? Как он отражает сущности реального мира?

Как реализованы наследование, инкапсуляция и полиморфизм в Ruby?

Для чего используются модули и примеси (mixins)?

Задание:

Напишите пример кода, использующий все три особенности ООП: наследование, инкапсуляцию и полиморфизм.

Тема 5. Тестирование кода

Вопросы:

Зачем нужно автоматизированное тестирование?

Какие существуют подходы к написанию тестов (TDD, BDD)?

Какие инструменты тестирования используются в Ruby и Rails?

Задание:

Напишите пример автоматического теста функции, используя Cucumber или RSpec.

Тема 6. Знакомство с Ruby On Rails

Вопросы:

Что такое web-фреймворк? Какие задачи он решает?

Почему появился Ruby on Rails и в чём его преимущества?

Задание:

Установите Ruby on Rails на свой компьютер и создайте первое приложение.

Тема 7. Основы Ruby On Rails

Вопросы:

Какие основные составляющие имеет фреймворк Ruby on Rails?

Как устроена структура Rails-приложения?

Для чего используются генераторы и scaffolding?

Задание:

Создайте новое шаблонное приложение и опишите его структуру.

Тема 8. Rails и базы данных

Вопросы:

Как организовано взаимодействие Rails-приложения с СУБД?

Что такое ActiveRecord, миграции, и как они применяются?

Приведите пример SQL-запроса, сгенерированного через ActiveRecord.

Задание:

Создайте новую базу данных PostgreSQL и подключите её к своему проекту.

Тема 9. Rails Models

Вопросы:

Как модель в Rails отражает данные в СУБД?

Какие типы связей между моделями существуют (has_one, has_many, belongs_to)?

Как реализовать полиморфизм в связях моделей?

Задание:

Создайте модели с взаимосвязями, описанными на лекции.

Тема 10. Rails Controllers

Вопросы:

Какую роль играют контроллеры в архитектуре MVC?

Как реализуются CRUD-операции в контроллерах?

Как обрабатываются параметры запросов?

Задание:

Реализуйте контроллер, выполняющий поиск объектов по заданному параметру.

Тема 11. Rails Views

Вопросы:

Для чего используются шаблоны (views) в Rails?

Что такое ERB, helpers, assets pipeline?

Как подключить Bootstrap к Rails-приложению?

Задание:

Оформите представление своего проекта с использованием Bootstrap.

Тема 12. Rails Views and JS

Вопросы:

Как интегрировать JavaScript и Ajax в Rails-приложение?

Приведите пример использования Ajax в представлении Rails.

Задание:

Переделайте представление проекта для работы с Ajax-запросами.

Тема 13. i18n и l10n

Вопросы:

Как реализуется локализация в Rails-приложениях?

Какие файлы и методы используются для перевода интерфейса?

Задание:

Локализируйте на русский язык представления своего проекта.

Тема 14. Хостинг Rails-приложений

Вопросы:

Какие существуют платформы для хостинга Rails-приложений?

Как осуществляется развертывание приложения на heroku.com?

Задание:

Выложите своё приложение на heroku.com и предоставьте ссылку преподавателю.

2. Вопросы и задания для промежуточной аттестации (зачёт)

Теоретические вопросы:

Опишите архитектуру MVC и её реализацию в Ruby on Rails.

Объясните процесс создания и миграции базы данных в Rails.

Расскажите о принципах работы ActiveRecord.

Перечислите основные этапы жизненного цикла Rails-приложения.

Объясните, как реализуется безопасность в Rails-приложениях.

Практические задания:

Разработайте небольшое приложение на Ruby on Rails, реализующее простую предметную область (например,

библиотека, блог, список задач).

Реализуйте модели, контроллеры и представления для CRUD-операций.

Добавьте в приложение автоматизированные тесты для одной из моделей или контроллеров.

Разместите приложение на heroku.com и предоставьте ссылку.

Аналитические вопросы:

Проанализируйте преимущества и недостатки использования Ruby on Rails для web-разработки.

Предложите способы оптимизации производительности Rails-приложения.

Опишите возможные уязвимости и методы их предотвращения в web-приложениях.

3. Вопросы для экзамена (промежуточная аттестация)

1. Язык Ruby и основы ООП

Опишите основные синтаксические особенности языка Ruby.

Объясните понятие класса в Ruby и приведите пример его использования.

Как реализованы наследование, инкапсуляция и полиморфизм в Ruby?

Что такое блоки в Ruby и как они применяются?

Приведите пример динамической типизации в Ruby.

2. Контроль версий

Для чего используются системы контроля версий? Какие задачи решает Git?

Опишите процесс создания репозитория и работы с ветками в Git.

Как происходит совместная работа над проектом с использованием GitHub?

3. Тестирование программного кода

Для чего необходимо автоматизированное тестирование ПО?

В чем суть подходов TDD и BDD? Какие инструменты тестирования используются в Ruby/Rails?

Приведите пример автоматического теста для функции или класса.

4. Основы Ruby on Rails

Что такое web-фреймворк? В чем преимущества Ruby on Rails?

Опишите архитектуру MVC и её реализацию в Rails.

Как устроена структура Rails-приложения? Назовите основные директории и их назначение.

Какие команды генерации и scaffolding существуют в Rails?

5. Работа с базами данных

Как осуществляется взаимодействие Rails-приложения с СУБД?

Что такое ActiveRecord и миграции? Приведите пример миграции и SQL-запроса через ActiveRecord.

Какие типы связей между моделями существуют? Приведите примеры.

6. Rails Models, Controllers, Views

Как модель в Rails отражает данные в СУБД? Как реализуются связи между моделями?

Как реализовать CRUD-операции в контроллере Rails?

Как обрабатываются параметры запросов в контроллере?

Для чего используются шаблоны (views) в Rails? Что такое ERB, helpers, assets pipeline?

7. Интеграция с внешними технологиями

Как интегрировать Bootstrap и JavaScript в Rails-приложение?

Как реализовать Ajax-запросы и обновление данных без перезагрузки страницы?

8. Локализация и хостинг

Как реализовать локализацию (i18n, l10n) в Rails-приложении?

Опишите процесс развертывания приложения на heroku.com.

9. Безопасность и оптимизация

Какие основные угрозы безопасности существуют для web-приложений? Как их предотвращать в Rails?

Какие способы оптимизации производительности Rails-приложения вы знаете?

Практические задания (на экзамене)

Разработайте небольшое приложение на Ruby on Rails (например, блог, библиотека, список задач), реализующее:

Модели с взаимосвязями (has_many, belongs_to);

CRUD-операции через контроллеры;

Представления с использованием ERB и Bootstrap;

Одну или несколько форм с валидацией;

Пример локализации интерфейса;

Пример автоматического теста (RSpec или Cucumber).

Разместите приложение на heroku.com и предоставьте ссылку преподавателю.

Проанализируйте архитектуру своего приложения:

Опишите, как реализованы принципы MVC;

Какие методы обеспечения безопасности использованы;

Как организовано взаимодействие с базой данных.

Перечень вопросов к зачёту

Основы синтаксиса языка Ruby

Какие базовые конструкции языка Ruby вы знаете?

Каковы особенности синтаксиса Ruby?

Синтаксические конструкции, блоки, передача аргументов, базовые типы данных

Как реализованы блоки в Ruby?

Как осуществляется передача аргументов в методы?

Какие типы данных используются в Ruby?

Приведите пример динамической типизации.

Системы контроля версий (Git)

Для чего нужны системы контроля версий?
Какие задачи решает использование системы контроля версий?
Каковы основные команды Git?
Классы и ООП в Ruby
Что такое класс в Ruby?
Как реализованы наследование, полиморфизм и инкапсуляция в Ruby?
Как класс отражает сущность реального мира?
Автоматизированное тестирование программного обеспечения
Для чего необходимо автоматизированное тестирование?
Какие проблемы решает автоматизация тестов?
Какие существуют подходы к написанию тестов (TDD, BDD)?
Какие инструменты тестирования используются в Ruby/Rails?
Web-фреймворки и Ruby on Rails
Что такое web-фреймворк?
Какова краткая история возникновения web-фреймворков?
Какие задачи облегчает использование фреймворка для разработчика?
Архитектура и основные модули Ruby on Rails

Каковы основные составляющие фреймворка Ruby on Rails?
Какие функции выполняет каждый модуль Rails?
Взаимодействие Rails-приложения с базами данных
Как Rails-приложение взаимодействует с СУБД?
Что такое ActiveRecord и каковы его основные методы?
Приведите примеры составления запросов к базе данных.
Модели в Rails
Что такое модель в Rails?
Как модель отражает данные в СУБД?
Какие данные хранит модель?
Какова сфера ответственности класса-модели?
Контроллеры в Rails
Какова роль контроллеров в Rails-приложении?
Как реализуется логика приложения в контроллерах?
Как осуществляется обработка запросов?
Представления (views) в Rails
Для чего используются представления в Rails?
Что такое шаблоны ERB?
Как используются helpers и assets pipeline?
Как интегрируется Bootstrap в представления?
Интеграция JavaScript и Ajax в Rails
Как интегрируется JavaScript в Rails-приложение?
Как реализуются Ajax-запросы в представлениях Rails?
Локализация и интернационализация (i18n, l10n) в Rails
Как реализуется локализация интерфейса в Rails-приложениях?
Какие инструменты и файлы используются для перевода?
Хостинг Rails-приложений
Какие существуют платформы для размещения Rails-приложений?
Как осуществляется деплой приложения на heroku.com?

Перечень вопросов к экзамену

Раздел 1. Язык Ruby и основы программирования

Краткая история языка Ruby. Почему именно Ruby?

Основы синтаксиса языка Ruby, базовые конструкции (переменные, операторы, массивы, хэши, итерации).

Принцип «Всё - объекты» в Ruby.

Области видимости и приоритеты операторов в Ruby.

Синтаксические конструкции, блоки, передача аргументов, базовые типы данных.

Динамическая типизация в Ruby: примеры и преимущества.

Системы контроля версий: назначение, основные понятия, преимущества использования Git.

Основные команды Git, ветвления, слияния, работа с GitHub, pull-requests.

Понятие класса в Ruby: структура, методы, поля, конструкторы.

Принципы ООП в Ruby: инкапсуляция, наследование, полиморфизм, их реализация.

Модули и примеси (mixins) в Ruby, интерфейс объекта.

Автоматизированное тестирование: зачем нужно, какие задачи решает.

Подходы к тестированию: TDD, BDD, инструменты RSpec и Cucumber.

Пример автоматического теста функции на Ruby.

Раздел 2. Основы Ruby on Rails и разработка web-приложений

Краткая история web-фреймворков. Что такое web-фреймворк?

Преимущества и особенности Ruby on Rails.

Необходимость разделения данных, логики и представления (MVC).

Основные составляющие и модули Ruby on Rails, функции каждого модуля.

Создание нового Rails-приложения: структура, основные директории, генераторы, scaffolding.

СУБД: отличие SQL и NoSQL, выбор СУБД для Rails-приложения.

Установка и настройка PostgreSQL, утилиты psql, createdb, createuser.

ActiveRecord: назначение, основные методы, миграции, примеры SQL-запросов через ORM.

Модель в Rails: как класс, как отражение данных в СУБД, ответственность класса.

Взаимосвязи между моделями: has_one, has_many, belongs_to, полиморфизм в связях моделей.

Валидация данных в моделях Rails.

Контроллеры в Rails: реализация логики приложения, основы обработки запросов.

CRUD и REST: маршрутизация, работа с параметрами, примеры кода контроллера.

Представления (views) в Rails: шаблоны ERB, helpers, assets pipeline.

Интеграция Bootstrap: назначение, способы подключения, оформление интерфейса.

Интеграция JavaScript и Ajax в Rails-приложениях: назначение, примеры реализации.

Принципы работы Ajax, примеры асинхронных запросов в Rails.

Локализация и интернационализация (i18n, l10n) в Rails: задачи, настройка языковых файлов.

Хостинг Rails-приложений: обзор платформ, этапы деплоя на heroku.com.

Особенности публикации приложения на Heroku, настройка окружения.

Безопасность web-приложений на Ruby on Rails: типовые угрозы, методы защиты.

Оптимизация производительности Rails-приложения: кеширование, оптимизация запросов.

Современные подходы к тестированию, отладке и сопровождению Rails-приложений.

Внедрение и адаптация Rails-приложений в бизнес-процессы организации.

Оценка экономических затрат на проекты информатизации с использованием Rails.

Влияние open source-сообщества на развитие Ruby и Rails.

Примерные практические задания:

Написать пример класса в Ruby с инкапсуляцией, наследованием и полиморфизмом.

Реализовать миграцию и модель с ассоциациями в Rails.

Создать контроллер с CRUD-операциями и реализовать поиск по базе данных.

Оформить представление с использованием Bootstrap и реализовать Ajax-запрос.

Настроить локализацию для одной из страниц приложения.

Описать процесс деплоя приложения на Heroku.

Комплект контрольных вопросов для устных опросов

1. Знакомство с Ruby

Каковы основные синтаксические особенности языка Ruby?

Какие базовые конструкции языка Ruby вы знаете? Приведите примеры.

2. Основы Ruby

Что означает принцип «Всё - объекты» в Ruby?

Каковы основные типы данных в Ruby?

Как реализованы блоки и итераторы в Ruby?

Как осуществляется передача аргументов в методы?

Приведите пример динамической типизации в Ruby.

3. Контроль версий с GIT

Для чего используются системы контроля версий?

Какие задачи решает использование системы контроля версий в командной разработке?

Назовите основные команды Git и их назначение.

Как осуществляется работа с ветками и слияниями в Git?

Как происходит взаимодействие с GitHub?

4. Классы и ООП в Ruby

Что такое класс в Ruby? Как класс отражает сущность реального мира?

Как реализованы наследование, инкапсуляция и полиморфизм в Ruby?

Для чего используются модули и примеси (mixins) в Ruby?

Приведите пример кода, реализующий основные принципы ООП.

5. Тестирование кода

Для чего необходимо автоматизированное тестирование программного обеспечения?

Какие задачи решает автоматизация тестов?

Какие существуют подходы к написанию тестов (TDD, BDD)?

Какие инструменты тестирования используются в Ruby и Rails?

6. Знакомство с Ruby On Rails

Что такое web-фреймворк? Какова история появления web-фреймворков?

Какие задачи облегчает использование фреймворка для разработчика?

7. Основы Ruby On Rails

Какие основные составляющие имеет фреймворк Ruby On Rails?

Какие функции выполняет каждый модуль Rails?

Какова структура Rails-приложения?

8. Rails и базы данных

Как Rails-приложение взаимодействует с СУБД?

Что такое ActiveRecord и миграции?

Приведите примеры составления запросов к базе данных через ActiveRecord.

9. Rails Models

Что такое модель в Rails? Как модель отражает данные в СУБД?

Какие типы связей между моделями существуют (has_one, has_many, belongs_to)?

Как реализовать полиморфизм в связях моделей?

Какие данные и функции находятся в сфере ответственности класса-модели?

10. Rails Controllers

Какова роль контроллеров в архитектуре MVC?

Как реализуются CRUD-операции в контроллерах Rails?

Как осуществляется обработка параметров запросов в контроллере?

Приведите пример кода контроллера.

11. Rails Views

Для чего используются представления (views) в Rails?

Что такое шаблоны ERB? Как они работают?

Как используются helpers и assets pipeline?
Как подключить и использовать Bootstrap в Rails-приложении?
12. Rails Views и JavaScript (Ajax)
Как интегрировать JavaScript и Ajax в Rails-приложение?
Приведите пример использования Ajax в представлении Rails.
13. Локализация и интернационализация (i18n, l10n)
Как реализуется локализация интерфейса в Rails-приложениях?
Какие файлы и методы используются для перевода интерфейса?
14. Хостинг Rails-приложений
Какие существуют платформы для размещения Rails-приложений?
Как осуществляется развертывание приложения на heroku.com?
Какие этапы включает деплой Rails-приложения?

Комплект заданий для лабораторных работ

Лабораторная работа 1. Основы синтаксиса Ruby

Цели:

Ознакомиться с базовыми конструкциями языка Ruby.

Научиться писать простые программы на Ruby.

Пошаговая инструкция:

Изучить синтаксис языка Ruby: переменные, типы данных, операторы.

Написать программу, которая выводит на экран текст и результат арифметических операций.

Создать и использовать массивы и хэши.

Реализовать простой цикл с использованием итераторов.

Критерии оценки:

Корректность синтаксиса и работоспособность программы.

Использование базовых конструкций языка.

Чистота и читаемость кода.

Лабораторная работа 2. Работа с классами и ООП в Ruby

Цели:

Освоить принципы объектно-ориентированного программирования в Ruby.

Научиться создавать классы, методы, использовать наследование.

Пошаговая инструкция:

Создать класс с атрибутами и методами.

Реализовать наследование и переопределение методов.

Использовать инкапсуляцию (private, protected методы).

Продемонстрировать полиморфизм на примере нескольких классов.

Критерии оценки:

Правильная реализация классов и наследования.

Корректное использование модификаторов доступа.

Соответствие принципам ООП.

Лабораторная работа 3. Контроль версий с Git

Цели:

Научиться работать с системой контроля версий Git.

Освоить основные команды и работу с удалённым репозиторием.

Пошаговая инструкция:

Установить Git и создать локальный репозиторий.

Сделать несколько коммитов с изменениями в проекте.

Создать ветку, внести изменения, выполнить слияние веток.

Опубликовать репозиторий на GitHub и сделать pull-request.

Критерии оценки:

Правильность выполнения команд Git.

Наличие истории коммитов с понятными сообщениями.

Успешное создание и слияние веток.

Лабораторная работа 4. Тестирование кода с RSpec/Cucumber

Цели:

Познакомиться с автоматизированным тестированием в Ruby.

Научиться писать тесты и запускать их.

Пошаговая инструкция:

Установить RSpec и/или Cucumber в проект.

Написать тесты для простых функций или классов.

Запустить тесты и проанализировать результаты.

Исправить ошибки, если тесты не проходят.

Критерии оценки:

Наличие корректно написанных тестов.

Успешное прохождение тестов.

Понимание принципов TDD/BDD.

Лабораторная работа 5. Создание базового Rails-приложения

Цели:

Освоить создание и структуру Rails-приложения.

Запустить сервер и проверить работу приложения.

Пошаговая инструкция:

Установить Ruby on Rails.

Создать новое приложение командой rails new.

Ознакомиться со структурой каталогов.

Запустить сервер и открыть стартовую страницу.

Критерии оценки:

Корректное создание приложения.

Запуск сервера без ошибок.

Понимание структуры проекта.

Лабораторная работа 6. Работа с базами данных и миграциями

Цели:

Научиться создавать и подключать базу данных PostgreSQL к Rails-приложению.

Освоить миграции для создания таблиц.

Пошаговая инструкция:

Установить и настроить PostgreSQL.

Создать базу данных и пользователя.

Настроить подключение в database.yml.

Создать миграцию для таблицы, выполнить миграцию.

Проверить структуру базы.

Критерии оценки:

Корректное создание и подключение БД.

Успешное выполнение миграций.

Соответствие структуры БД требованиям.

Лабораторная работа 7. Создание моделей и связей

Цели:

Научиться создавать модели и устанавливать связи между ними.

Реализовать валидации данных.

Пошаговая инструкция:

Создать модели с помощью генераторов.

Определить связи has_many, belongs_to.

Добавить валидации (например, присутствие, уникальность).

Проверить работу моделей через консоль Rails.

Критерии оценки:

Правильное определение моделей и связей.

Корректная работа валидаций.

Отсутствие ошибок при работе с моделями.

Лабораторная работа 8. Разработка контроллеров и маршрутизация

Цели:

Освоить создание контроллеров и маршрутов в Rails.

Реализовать CRUD-операции.

Пошаговая инструкция:

Создать контроллер с действиями index, show, new, create, edit, update, destroy.

Настроить маршруты в routes.rb.

Реализовать обработку параметров и логику действий.

Проверить работу через браузер.

Критерии оценки:

Полнота и корректность CRUD-операций.

Правильная маршрутизация.

Отсутствие ошибок при выполнении запросов.

Лабораторная работа 9. Создание представлений и оформление с Bootstrap

Цели:

Научиться создавать представления с использованием ERB.

Подключить Bootstrap для стилизации.

Пошаговая инструкция:

Создать шаблоны представлений для основных действий контроллера.

Подключить Bootstrap через CDN или gem.

Оформить страницы с использованием классов Bootstrap.

Проверить отображение страниц.

Критерии оценки:

Корректное отображение данных.

Использование Bootstrap для оформления.

Чистота и удобство интерфейса.

Лабораторная работа 10. Интеграция Ajax и JavaScript

Цели:

Освоить асинхронные запросы в Rails.

Реализовать динамическое обновление страниц.

Пошаговая инструкция:

Добавить JavaScript и Ajax в представления.

Реализовать обновление части страницы без перезагрузки.

Проверить работу функционала.

Критерии оценки:

Корректная работа Ajax-запросов.

Отсутствие ошибок в консоли браузера.

Улучшение пользовательского опыта.

Лабораторная работа 11. Локализация интерфейса

Цели:

Научиться использовать i18n для локализации.

Реализовать переключение языков.

Пошаговая инструкция:

Настроить файлы локализации.

Перевести основные элементы интерфейса.

Добавить переключатель языка.

Проверить корректность отображения.

Критерии оценки:

Полнота перевода.

Корректная работа переключения языков.

Отсутствие ошибок.

Лабораторная работа 12. Деплой приложения на Heroku

Цели:

Освоить процесс публикации Rails-приложения в облаке.

Проверить работу приложения в продакшн-среде.

Пошаговая инструкция:

Создать аккаунт на Heroku.

Подготовить приложение к деплою (Gemfile, конфигурация БД).

Выполнить деплой через Git.

Проверить работу приложения по ссылке.

Комплект тем для самостоятельной работы

1. Знакомство с Ruby

Задание: Изучить основы синтаксиса языка Ruby, основные базовые конструкции языка.

Вопросы для самопроверки:

Какие основные конструкции языка Ruby вы знаете?

В чём особенности синтаксиса Ruby?

2. Основы Ruby

Задание: Освоить синтаксические конструкции, блоки, передачу аргументов, базовые типы данных. Привести примеры динамической типизации.

Вопросы для самопроверки:

Как реализованы блоки в Ruby?

Как осуществляется передача аргументов в методы?

Какие типы данных используются в Ruby?

Приведите пример динамической типизации.

3. Контроль версий с GIT

Задание: Изучить назначение и основные принципы работы системы контроля версий Git, освоить базовые команды.

Вопросы для самопроверки:

Для чего нужны системы контроля версий?

Какие задачи решает использование системы контроля версий?

4. Классы, ООП

Задание: Изучить понятие класса как описания сущности реального мира, освоить наследование, полиморфизм и инкапсуляцию в Ruby.

Вопросы для самопроверки:

Что такое класс в Ruby?

Как реализованы наследование, полиморфизм и инкапсуляция в Ruby?

5. Тестирование кода

Задание: Изучить основы автоматизированного тестирования программного обеспечения, подходы к написанию тестов

(TDD, BDD), инструменты тестирования.

Вопросы для самопроверки:

Какие задачи решает автоматизация тестов?

Какие существуют подходы к написанию тестов?

6. Знакомство с Ruby On Rails

Задание: Подготовиться к контрольной работе: изучить понятие web-фреймворка, историю возникновения, задачи, которые облегчает фреймворк для разработчика.

Вопросы для самопроверки:

Что такое web-фреймворк?

Какие задачи решает использование фреймворка?

7. Основы Ruby On Rails

Задание: Изучить основные составляющие фреймворка Ruby On Rails, назначение и функции модулей.

Вопросы для самопроверки:

Какие основные модули входят в Rails?

Каковы функции каждого модуля?

8. Rails и базы данных

Задание: Изучить способы взаимодействия Rails-приложения с СУБД, методы ActiveRecord, примеры запросов к базе данных.

Вопросы для самопроверки:

Как Rails-приложение взаимодействует с СУБД?

Какие основные методы ActiveRecord вы знаете?

9. Rails Models

Задание: Изучить понятие модели как класса и отражения данных в СУБД, определить сферу ответственности класса-модели.

Вопросы для самопроверки:

Какие данные хранит модель?

Какова ответственность класса-модели?

10. Rails Controllers

Задание: Изучить роль контроллеров в архитектуре Rails, способы реализации логики приложения, обработку запросов.

Вопросы для самопроверки:

Как реализовать контроллер в Rails?

Как осуществляется обработка параметров запроса?

11. Rails Views

Задание: Подготовиться к контрольной работе: изучить назначение представлений, шаблоны ERB, helpers, assets pipeline, интеграцию Bootstrap.

Вопросы для самопроверки:

Для чего нужны представления в Rails?

Как используются helpers и assets pipeline?

12. Rails Views and JS

Задание: Изучить интеграцию JavaScript и Ajax в Rails-приложение, примеры использования Ajax-запросов.

Вопросы для самопроверки:

Как интегрируется JavaScript в Rails?

Как реализовать Ajax-запрос в представлении?

13. i18n и I18n

Задание: Изучить методы локализации интерфейса Rails-приложения, настройку языковых файлов.

Вопросы для самопроверки:

Как реализуется локализация в Rails?

Какие файлы используются для перевода?

14. Хостинг Rails-приложений

Задание: Изучить платформы для размещения Rails-приложений, этапы деплоя на heroku.com.

Вопросы для самопроверки:

Как осуществляется деплой приложения на Heroku?

Какие этапы включает публикация приложения?

Общие рекомендации

Для каждой темы подготовьте краткий конспект и выполните практические задания (написание кода, настройка среды, разбор примеров).

По итогам самостоятельной работы подготовьте отчет или ответьте на вопросы для самопроверки.

Часть заданий предполагает публикацию кода на GitHub и/или деплой приложения на Heroku.

Формы контроля:

Проверка домашних заданий и отчетов

Контрольные работы

Защита мини-проектов

Устные опросы и тестирование

Комплект тестовых заданий

Уровень: ЗНАТЬ

1. Какой из перечисленных языков программирования лежит в основе фреймворка Ruby on Rails?

A) Python

B) Ruby

C) Java

D) PHP

Ответ: B

2. Как называется принцип ООП, при котором подкласс наследует свойства и методы суперкласса?

A) Инкапсуляция

B) Полиморфизм

C) Наследование

D) Абстракция

Ответ: C

3. Какой тип данных в Ruby позволяет хранить пары "ключ-значение"?

A) Array

B) Hash

C) String

D) Set

Ответ: B

4. Какой командой в Git создаётся новая ветка?

A) git merge

B) git branch

C) git clone

D) git pull

Ответ: B

5. Какой файл отвечает за описание зависимостей в Rails-проекте?

A) config.ru

B) Gemfile

C) package.json

D) requirements.txt

Ответ: B

6. Какой шаблонизатор по умолчанию используется в Rails для представлений?

A) Haml

B) ERB

C) Slim

D) Jade

Ответ: B

7. Какой метод ActiveRecord используется для поиска записи по id?

A) find_by

B) locate

C) find

D) search

Ответ: C

Уровень: ЗНАТЬ, УМЕТЬ

8. Какой командой создаётся новое Rails-приложение?

A) rails generate app

B) rails new

C) rails create

D) rails init

Ответ: B

9. Какой командой выполняется миграция в Rails?

A) rails migrate

B) rake db:migrate

C) rails db:create

D) rails db:rollback

Ответ: B

10. Какой метод используется для валидации присутствия значения в модели Rails?

A) validates_presence_of

B) validate_presence

C) require

D) validates

Ответ: A

11. Какой метод контроллера Rails отвечает за отображение списка всех записей?

A) show

B) index

C) list

D) display

Ответ: B

12. Какой файл отвечает за маршрутизацию в Rails-приложении?

- A) routes.rb
- B) paths.rb
- C) navigation.rb
- D) config.rb

Ответ: A

13. Какой инструмент используется для написания модульных тестов в Ruby?

- A) JUnit
- B) RSpec
- C) Mocha
- D) Selenium

Ответ: B

14. Какой метод используется для передачи данных из контроллера во view в Rails?

- A) set
- B) assign
- C) instance variable (@variable)

D) pass

Ответ: C

Уровень: ЗНАТЬ, УМЕТЬ, ВЛАДЕТЬ

15. Какой командой можно развернуть Rails-приложение на платформе Heroku?

- A) heroku deploy
- B) git push heroku master
- C) rails deploy heroku
- D) heroku start

Ответ: B

16. Какой способ интеграции позволяет обновлять часть страницы без полной перезагрузки?

- A) WebSockets
- B) Ajax
- C) REST
- D) SOAP

Ответ: B

17. Какой механизм Rails используется для локализации интерфейса?

- A) i18n
- B) l10n
- C) translate
- D) localize

Ответ: A

18. Какой тип связи между моделями описывает отношение "один ко многим"?

- A) has_one
- B) has_many
- C) belongs_to
- D) has_and_belongs_to_many

Ответ: B

19. Какой метод используется для защиты от CSRF-атак в Rails-контроллере?

- A) protect_from_forgery
- B) csrf_protect
- C) secure_form
- D) verify_authenticity

Ответ: A

20. Какой инструмент Rails позволяет создавать шаблонные структуры кода для моделей,

контроллеров и представлений?

- A) rails generator
- B) rails scaffolding
- C) rails builder
- D) rails template

Ответ: B

Кейс-задания

Кейс 1. Разработка мини-CMS для новостного сайта

Ситуация:

Компания планирует создать новостной сайт с возможностью публикации статей, их редактирования и удаления, а также с разделением пользователей на редакторов и читателей.

Задание:

Проанализируйте предметную область и определите основные сущности (модели) и их связи.

Разработайте структуру базы данных (модели, миграции) для хранения статей, пользователей и ролей.

Реализуйте CRUD-интерфейс для статей с разграничением прав доступа (редактор/читатель).

Организируйте регистрацию и аутентификацию пользователей.

Реализуйте автоматическое тестирование основных функций.

Выложите приложение на Heroku и предоставьте ссылку.

Кейс 2. Интеграция внешнего API в Rails-приложение

Ситуация:

Вам поручено добавить в существующее Rails-приложение функционал отображения текущей погоды в городе

пользователя, используя открытое API (например, OpenWeatherMap).

Задание:

Изучите документацию выбранного API.

Реализуйте сервисный объект или модуль для получения данных о погоде по заданному городу.

Добавьте на главную страницу приложения виджет, отображающий актуальную погоду.

Обеспечьте обработку ошибок при недоступности API.

Оформите интерфейс с помощью Bootstrap.

Кейс 3. Безопасность и аудит Rails-приложения

Ситуация:

В процессе аудита вашего Rails-приложения обнаружены потенциальные уязвимости: отсутствие защиты от CSRF, XSS

и SQL-инъекций.

Задание:

Проанализируйте код приложения и выявите уязвимые места.

Реализуйте защиту от CSRF-атак средствами Rails.

Проверьте шаблоны на предмет XSS и обеспечьте экранирование пользовательского ввода.

Проверьте использование параметризованных запросов для защиты от SQL-инъекций.

Подготовьте отчёт по проведённому аудиту и реализованным мерам.

Кейс 4. Локализация и интернационализация приложения

Ситуация:

Компания выходит на международный рынок и требует, чтобы ваше приложение поддерживало как минимум русский и английский языки.

Задание:

Настройте систему i18n в Rails-приложении.

Переведите все основные элементы интерфейса на английский и русский языки.

Реализуйте переключение языка интерфейса пользователем.

Протестируйте корректность отображения информации на обоих языках.

Оформите краткую инструкцию для пользователей по смене языка.

Кейс 5. Оценка экономической эффективности внедрения Rails-приложения

Ситуация:

Руководство компании просит подготовить обоснование экономической целесообразности внедрения автоматизированной системы учёта заявок на обслуживание (на базе Rails).

Задание:

Опишите бизнес-процесс до и после внедрения системы.

Рассчитайте основные затраты на разработку, внедрение и сопровождение (человеко-часы, хостинг, поддержка).

Оцените ожидаемые выгоды: сокращение времени обработки заявок, снижение числа ошибок, повышение прозрачности.

Составьте сравнительную таблицу "затраты – выгоды".

Подготовьте презентацию для руководства.

Кейс 6. Реинжиниринг бизнес-процесса с использованием Rails

Ситуация:

В отделе кадров устаревшая система учёта отпусков. Необходимо автоматизировать процесс подачи, согласования и учёта заявок на отпуск.

Задание:

Проанализируйте текущий процесс (AS IS), выявите основные проблемы.

Предложите новую модель бизнес-процесса (TO BE) с использованием Rails-приложения.

Разработайте прототип приложения: модели, контроллеры, маршруты.

Реализуйте базовый интерфейс для подачи и согласования заявок.

Подготовьте отчёт по реинжинирингу и демонстрацию прототипа.

Рекомендации по использованию кейс-заданий

Кейс-задания могут использоваться для текущего контроля, промежуточной аттестации, защиты проектов или в качестве самостоятельной работы.

Оценка проводится по критериям: полнота анализа, качество реализации, оформление, защита.

Поощряется командная работа и использование современных инструментов (Git, CI/CD, Heroku и др.).

Темы письменных работ (эссе, рефераты, курсовые работы и др.)

1. История и эволюция языка программирования Ruby
2. Объектно-ориентированное программирование на Ruby: особенности и преимущества
3. Системы контроля версий в командной разработке: роль и преимущества Git
4. Автоматизированное тестирование в разработке программного обеспечения: подходы TDD и BDD
5. Архитектура MVC и её реализация во фреймворке Ruby on Rails

6. Сравнительный анализ фреймворков для веб-разработки: Ruby on Rails, Django, Laravel
7. Интеграция реляционных и нереляционных баз данных в веб-приложениях на Ruby on Rails
8. Роль моделей, контроллеров и представлений в структуре Rails-приложения
9. Интернационализация и локализация веб-приложений: опыт реализации в Ruby on Rails
10. Безопасность веб-приложений на Ruby on Rails: типовые угрозы и методы защиты
11. Использование Ajax и JavaScript в Rails-приложениях для повышения интерактивности
12. Деплой и сопровождение Rails-приложений: современные подходы и инструменты (Heroku, Docker, CI/CD)
13. Влияние open source-сообщества на развитие Ruby и Rails
14. Разработка и внедрение бизнес-процессов в Rails-приложениях: моделирование предметной области
15. Экономическая оценка проектов по информатизации на примере внедрения Rails-приложения