

Введение

Выполнение лабораторных работ предоставляет возможность использовать исходный код при разработке курсовой работы (с последующими доработками). Студентам необходимо изначально выбрать тему для клиент-серверного приложения. Также необходимо определить структуру хранения в базе данных. Вся кодовая база должна быть сохранена в GitLab.

Примерная тематика курсовых работ (проектов):

- разработать сервис для сохранения учетных данных пользователей
- разработать сервис авторизации пользователей
- разработать сервис для магазина
- разработать сервис для клиентов (сервис услуг)
- разработать сервис для системы бронирования

Лабораторная работа №1 «Базовые типы данных»

Подготовка:

1. необходимо установить среду разработки IntelliJ IDEA (Ultimate Edition)
2. необходимо создать Maven проект

Задание:

- создать структуру классов для дальнейшей разработки сервиса и хранения в базе данных (при разработке продумать возможность использования интерфейсов, наследования)
- реализовать хранение классов в структурах данных (списках, наборах, картах, очередях и т.д.) в зависимости от применения этих структур в дальнейшем
- реализовать алгоритмы обработки этих данных (циклы, логические операторы, преобразование строк и т.д.) – добавление новых объектов, поиск объектов по каким-либо полям, удаление объектов, вывод информации об объектах
- написать unit-тесты для реализованных алгоритмов (желательно использовать методологию TDD при разработке)
- при возможности, реализовать многопоточную обработку данных

Лабораторная работа №2 «Создание собственных классов»

Задание выполняется на основе кода из лабораторной работы №1.

Подготовка:

- скачать PostgreSQL server и PgAdmin (<https://www.postgresql.org/download/>)
- подключить к проекту (или создать новый проект) Spring Framework

Задание:

- реализовать схему базы данных в PgAdmin
- все дальнейшие реализации должны реализовывать модель Model-View-Controller, а также желательно использовать библиотеку Lombok
- доступ к данным осуществляется через Spring Data (JPA)
- реализовать работу с данными из кода и базой данных (реализация всех операций CRUD – создание объектов, вывод информации, обновление, удаление)
- реализовать REST-апи для операций из предыдущего пункта
- реализовать REST-апи для вывода информации из базы данных с использованием некоторых фильтров
- использовать @SpringBootTest для unit-тестов (тестовые методы не должны подключаться и каким-либо образом модифицировать базу данных – можно использовать Mockito)
- если в предыдущей лабораторной работе не реализована многопоточная обработка данных – реализовать в данной работе

Лабораторная работа №3 «Обработка исключений»

Задание выполняется на основе кода из лабораторной работы №2.

Задание:

- реализовать классы собственных исключений
- реализовать обработку исключений в коде (например, не должно быть единого catch для всех видов исключений, каждое должно

быть обработано и вынесено подробное описание ошибки и, если возможно, способ ее решения и т.п.)

Лабораторная работа №4 «Применение SpringFramework»

Задание выполняется на основе кода из лабораторной работы №3.

Задание:

- подключить Spring Security к проекту
- настроить пользователей и роли
- настроить методы REST-api для соответствующих ролей (например, операции DELETE может выполнить только пользователь с ролью ADMIN и т.д.)
- добавить логирование кода в консоль и файл (с использованием дефолтного Logback либо Log4j2 либо Slf4j)

Курсовая работа

Курсовая работа является результатом выполнения четырех лабораторных работ.

Требования:

- приложение требуется написать на языке Java с использованием фреймворка Spring и сборочной системы Maven
- должна быть подключена база данных PostgreSQL, для сохранения всех данных пользователей
- код должен быть покрыт unit-тестами (@SpringBootTest). При этом из тестов не должно быть реального обращения к какой-либо базе данных
- последовательность действий должна быть залогирована (на консоль и в файл)
- все исключительные ситуации должны быть обработаны
- на клиентской стороне используется командный интерфейс (можно реализовать и графический интерфейс)
- для работы с rest-api следует подключить swagger (при этом все методы должны быть сгруппированы, входные и выходные параметры подробно описаны, также описаны все возможные возвращаемые коды)