**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по производственной практике (по профилю специальности)**

По профессиональному модулю ПМ 11 - «Разработка, администрирование и защита баз данных»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студентка 3 курса 32919/4 группы

Кабачарова Анастасия Юрьевна

(Фамилия, имя, отчество)

Место прохождения практики: «Неофлекс» Санкт-Петербург, Белоостровская ул.,д.6, литера А, помещение 30-н

(наименование и адрес организации)

Период прохождения практики

с «1» апреля 2024 г. по «27» апреля 2024 г.

Руководитель практики

от учебной организации Коннова А.Е.

(подпись) (расшифровка подписи)

Руководитель с места

прохождения практики

(подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

М.П.

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на производственную практику (по профилю специальности)**

По профессиональному модулю ПМ 11 - «Разработка, администрирование и защита баз данных»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студентка 3 курса 32919/4 группы

Кабачарова Анастасия Юрьевна

(Фамилия, имя, отчество)

Место прохождения практики: «Неофлекс» Санкт-Петербург, Белоостровская ул.,д.6, литера А, помещение 30-н

(наименование и адрес организации)

Период прохождения практики

с «1» апреля 2024 г. по «27» апреля 2024 г.

**Виды работ, обязательные для выполнения** *(переносится из программы, соответствующего ПМ):*

1. Анализ предметной области
2. Проектирование базы-данных
3. Проектирование интерфейса информационной системы (приложение + база данных)
4. Тестирование информационной системы
5. Приемо-сдаточные мероприятия

Задание выдал «1» апреля 2024 г. \_\_\_ Коннова А.Е.

(подпись)  (Ф.И.О.)

Задание получил «1» апреля 2024 г. Кабачарова А. Ю. (подпись) (Ф.И.О.)

СОГЛАСОВАНО

Председатель ПЦК

\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения производственной практики**

**(по профилю специальности)**

По профессиональному модулю ПМ 11 - «Разработка, администрирование и защита баз данных»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студентка 3 курса 32919/4 группы

Кабачарова Анастасия Юрьевна (Фамилия, имя, отчество)

Место прохождения практики: «Неофлекс» Санкт-Петербург, Белоостровская ул.,д.6, литера А, помещение 30-н

(наименование и адрес организации)

Период прохождения практики

с «1» апреля 2024 г. по «27» апреля 2024 г.

Руководитель практики

от учебной организации Коннова А.Е.

(подпись) (расшифровка подписи)

Руководитель с места

прохождения практики

(подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

М.П.

Санкт-Петербург

2024

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 1.04.24 | Изучение разработки приложение |  |
| 2.04.24 |
| 3.04.24 | Разработка приложения-калькулятора |  |
| 8.04.24 |
| 9.04.24 | Изучения Swagger |  |
| 12.04.24 |
| 13.04.24 | Создание веб-инетерфейса для API |  |
| 18.04.24 |
| 19.04.24 | Написание тестов на проект |  |
| 25.04.24 | Сдача отчета |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc164692909)

[Задание № 1 «В среде Intellij IDEA Ultimate и используя Spring Framework Java написать клиент серверное приложение с микро сервисной архитектурой» 7](#_Toc164692910)

[Задание № 2 «Сервер должен принимать два аргумента и возвращать их сумму/разность.» 15](#_Toc164692911)

[Задание № 3 «Сгенерировать веб-интерфейс для ваших SpringBoot API с помощью Swagger 3.0.0» 18](#_Toc164692912)

[Задание №4 «Написать тесты на проект» 20](#_Toc164692913)

[ЗАКЛЮЧЕНИЕ 21](#_Toc164692914)

ВВЕДЕНИЕ

В данной практической работе я буду заниматься написанием клиент-серверного приложения на языке программирования Java с использованием Spring Framework. Так же ознакомлюсь с веб-интерфейсом Swagger и тестами.

Задание:

1. В среде Intellij IDEA Ultimate и используя Spring Framework Java написать клиент серверное приложение с микросервисной архитектурой.
2. Сервер должен принимать два аргумента и возвращать их сумму/разность.
3. Сгенерировать веб-интерфейс для ваших SpringBoot API с помощью Swagger 3.0.0.
4. \* Написать тесты на проект.

Задание № 1 «В среде Intellij IDEA Ultimate и используя Spring Framework Java написать клиент серверное приложение с микро сервисной архитектурой»

1. Скачать приложение Intellij IDEA Ultimate (Рисунок 1).

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 1 – загрузка приложения

1. Определение дирекции установки программы (Рисунок 2).

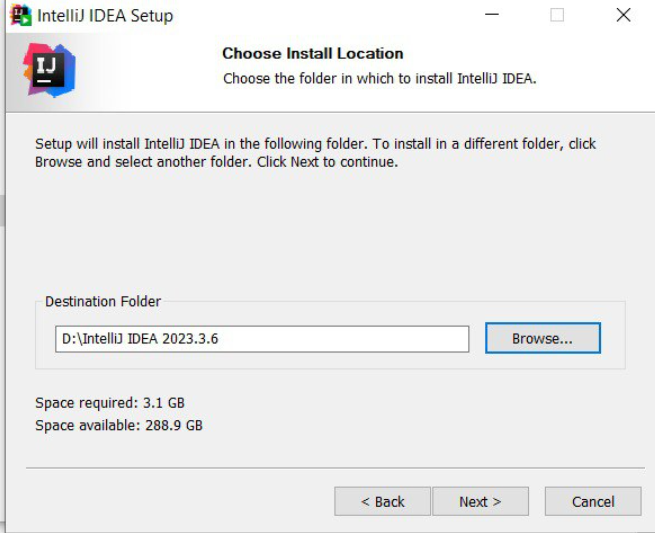


Рисунок 2 – выбор папки для установки

1. Параметры установки приложения: «Создание ярлыка на рабочем столе», а также создание ассоциации .java (Рисунок 3).

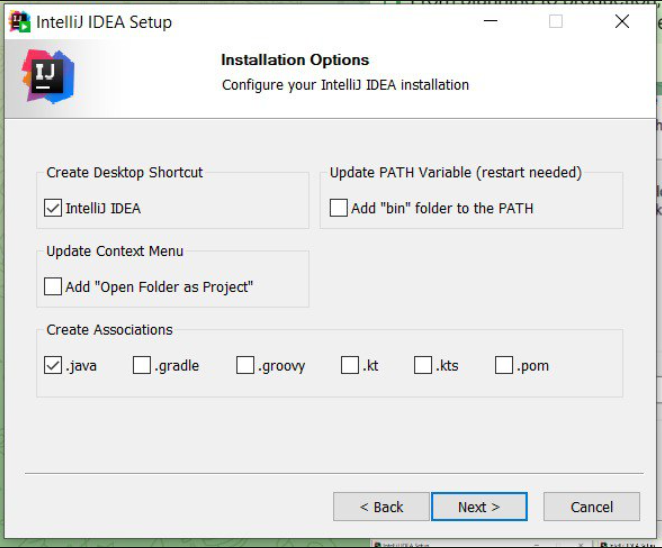


Рисунок 3 – конфигурация программы

1. Выбор папки, где будет располагаться файлы программы (Рисунок 4).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 4 -название папки

1. Установка программы на компьютер (Рисунок 5)

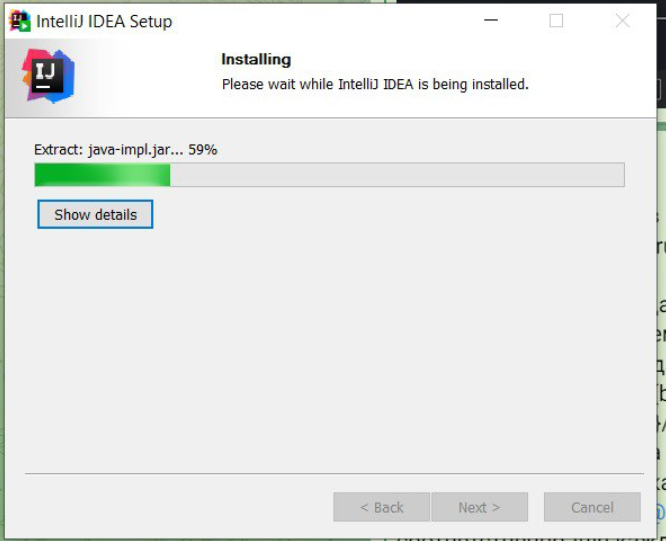


Рисунок 5 – процесс установки

1. Конфигурация проекта через сайт <https://start.spring.io/> (Рисунок 6).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 6 – конфигурация проекта

1. После того как выбрали основные настройки, заполняем описание проекта, выбор расширения и версии java (Рисунок 7).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 7 – описание проекта

1. Добавление зависимости Lombok – это библиотека, которая позволяет уменьшить количество шаблонного кода в классах за счет использования аннотаций, делая его более читаемым и поддерживаемым. А также зависимость Spring Web – это модуль фреймворка Spring, который обеспечивает поддержку веб-приложений в Java (Рисунок 8).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 8 – добавление зависимостей

Генерация конфигурации проекта при нажатии на кнопку Generate (Рисунок 9).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 9 - Генерация конфигурации проекта

1. Распаковать архив со сгенерированной конфигурации в папку с проектом в Intellij IDEA
2. Открытие проекта в Intellij IDEA Ultimate (Рисунок 10).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 10 – открытие проекта

1. Выбрать тип проекта Maven, он обеспечивает автоматизацию сборки проекта, управление зависимостями, тестирование, документирование. Maven использует файлы конфигурации для описания проекта и его зависимостей, что делает процесс разработки более структурированным и управляемым (Рисунок 11).

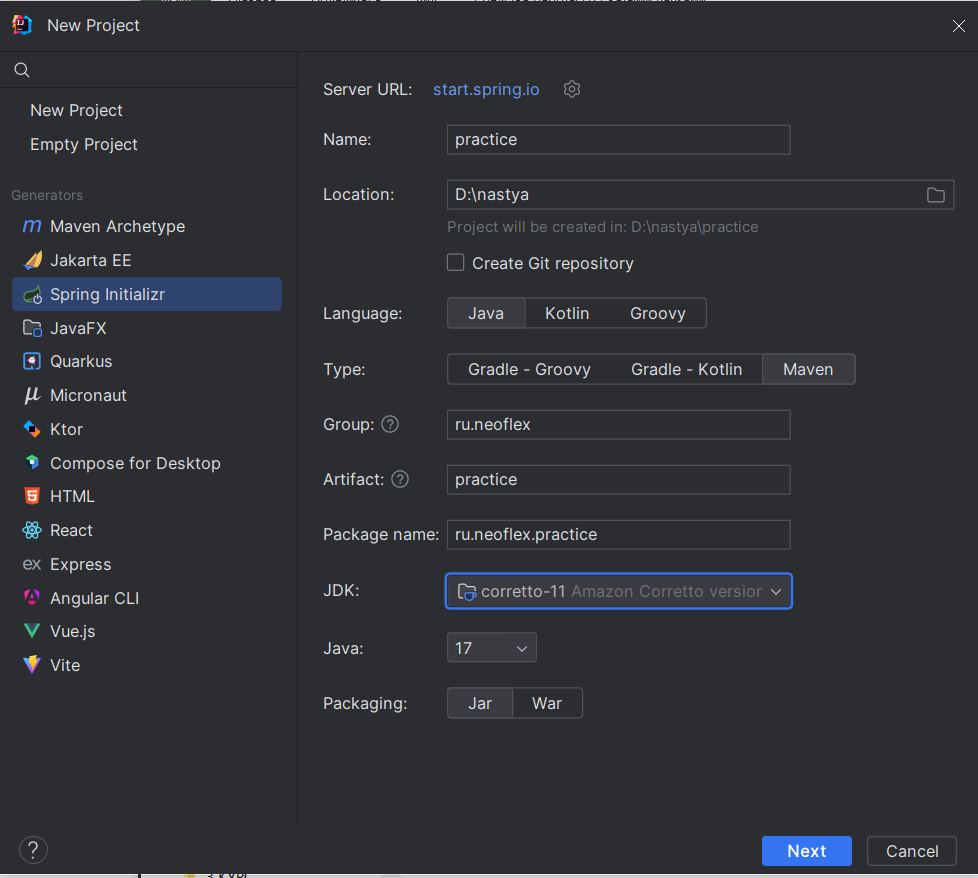


Рисунок 11 - настройки

1. Добавление Lombok – это библиотека, которая позволяет уменьшить количество шаблонного кода в классах за счет использования аннотаций, делая его более читаемым и поддерживаемым. Spring Web – это модуль фреймворка Spring, который обеспечивает поддержку веб-приложений в Java (Рисунок 12).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 12 – добавление зависимостей

1. В папке src/main/java/ru/neoflex/practice создать папку controller (Рисунок 13).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 13 – Создание папки controller

1. В папке (/controller) создаём файл [CalcController.java](https://calccontroller.java/) и далее работаем с проектом (Рисунок 14).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 14 – создание файла [CalcController.java](https://calccontroller.java/)

Задание № 2 «Сервер должен принимать два аргумента и возвращать их сумму/разность.»

1. Работа с проектом (Рисунок 15).

Изображение выглядит как снимок экрана, программное обеспечение, Мультимедийное программное обеспечение, текст

Автоматически созданное описание

Рисунок 15 – проект

1. Создание кода. Импорт необходимых классов из библиотеки фреймворка Spring. Аннотация @RestController указывает, что это класс контроллера, который обрабатывает HTTP-запросы и возвращает данные в формате JSON. Метод add с аннотацией @GetMapping("/plus/{a}/{b}") принимает два параметра a и b через пути URL и выполняет операцию сложения над ними, возвращая результат. Метод subtract с аннотацией @GetMapping("/minus/{a}/{b}") также принимает два параметра a и b через пути URL и выполняет операцию вычитания над ними, возвращая результат (Рисунок 16).

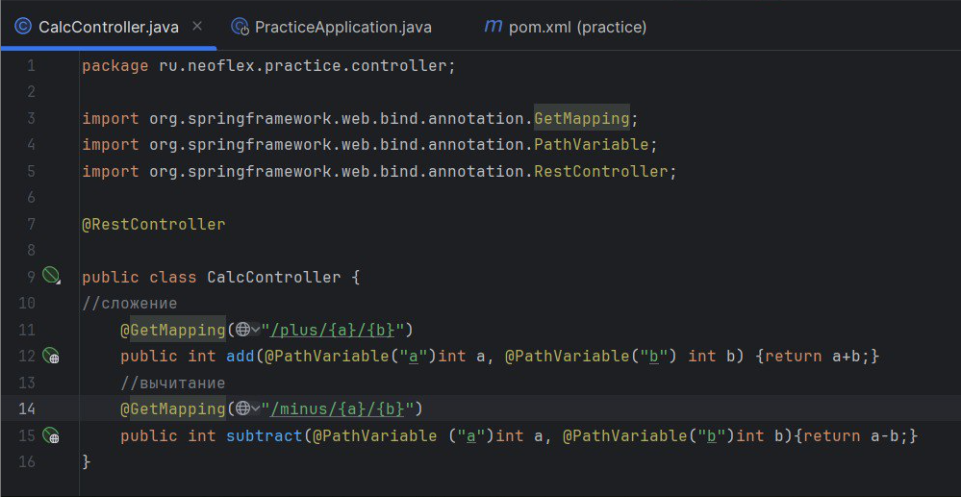


Рисунок 17 – код для калькулятора

1. Запустить приложение при нажатии на зеленую кнопку Run (Рисунок 18).

Изображение выглядит как текст, снимок экрана, Шрифт, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 18 – запуск приложения

1. Для того что бы протестировать метод сложения двух чисел необходимо в адресную строку веб-браузера ввести [http://localhost:8080/plus/1/8](http://localhost:8080/plus/1/8%20) (Рисунок 19).

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

Рисунок 19 – Результат метода сложения

1. Для того что бы протестировать метод разности двух чисел необходимо в адресную строку веб-браузера ввести <http://localhost:8080/minus/10/3> (Рисунок 20).

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рисунок 20 – Результат метода разности

Задание № 3 «Сгенерировать веб-интерфейс для ваших SpringBoot API с помощью Swagger 3.0.0»

1. Swagger — это набор инструментов для проектирования, документирования и тестирования веб-сервисов RESTful API. Он позволяет создавать спецификации API в формате OpenAPI, которые описывают структуру и функциональность API. Swagger также предоставляет интерфейс Swagger UI для взаимодействия с API и генерации документации на основе спецификации.
2. Для интеграции Swagger в проект необходимо добавить зависимость в файл pom.xml, выбрана версия 2.2.0 (Рисунок 21).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 21 – код в pom.xml

1. Обновление изменений в файле pom.xml при нажатии кнопки «Load Maven Changes» (Рисунок 22).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 22 – обновление изменений

1. Для доступа к веб-интерфейсу Swagger необходимо в браузере перейти по адресу <http://localhost:8080/swagger-ui/index.html>. Выбирается нужный метод из списка, заполняются поля и выводится результат.

Рисунок 23-вид вед-страницы

Рисунок 24 – результат метода разности

Рисунок 25 – результат метода сложения

Задание №4 «Написать тесты на проект»

1. Создание папки controller в директории src\test\java\ru.neoflex.practice (Рисунок 26).

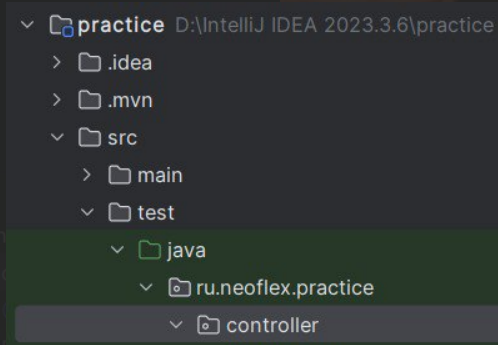


Рисунок 26 – создание папки controller

1. Создание в папке controller файл CalcTest.java (Рисунок 27).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 27 – создание файла CalcTest.java

1. В файле pom.xml. добавить зависимость (dependency), которая указывает на необходимость использования библиотеки JUnit Jupiter API версии 5.10.0 во время выполнения тестов. Зависимость указывает на группу (groupId) "org.junit.jupiter", артефакт (artifactId) "junit-jupiter-api" и указывает, что эта зависимость имеет область действия (scope) "test", что означает, что библиотека будет использоваться только во время выполнения тестов. (Рисунок 28).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 28 – зависимость dependency

1. В данном коде определены два метода testAdd и testSubtract, которые выполняют тестирование методов add и subtract соответственно. Метод testAdd принимает три параметра: a, b и expected. Он вызывает метод add из calcController с переданными значениями a и b, а затем сравнивает результат с ожидаемым значением expected при помощи метода assertEquals из класса Assertions.Метод testSubtract аналогичен методу testAdd, но тестирует метод subtract. В конце кода используются аннотации @Test для указания, что следующие методы должны быть выполнены как тесты. Внутри этих методов вызываются определенные тестовые случаи с различными значениями параметров для проверки корректности работы методов add и subtract (Рисунок 29).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 29 – код для теста

1. Тест считается успешным, если полученный результат совпадает с ожидаемым, в другом случае тест провален. Результат теста, который не пройден представлен на рисунке (Рисунок 30).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 30 – тест провален

ЗАКЛЮЧЕНИЕ

В данной практической работе я изучила базовый синтаксис языка программирования Java. Так же реализовала клиент-серверное приложение, которое принимает два числа от клиента и возвращает их в сумму или разность . Так же, я изучила способы взаимодействие клиента и сервера с помощью API используя Swagger 2.2.0, для удобного взаимодействия с кодом. Были написаны тесты для надежности функционала приложения.

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

По профессиональному модулю ПМ 11 - «Разработка, администрирование и защита баз данных»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студентка 3 курса 32919/4 группы

Кабачарова Анастасия Юрьевна

(Фамилия, имя, отчество)

Место прохождения практики: «Неофлекс» Санкт-Петербург, Белоостровская ул.,д.6, литера А, помещение 30-н

(наименование и адрес организации)

Период прохождения практики

с «1» апреля 2024 г. по «27» апреля 2024 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Виды выполненных работ обучающимся**  **во время практики** | **Объем работ, час.** | **Качество выполнения работ (оценка по пятибалльной системе)** |
| **Тема 1.** Анализ предметной области | 20 |  |
| **Тема 2**. Проектирование базы-данных | 35 |  |
| **Тема 3.** Проектирование интерфейса информационной системы (приложение + база данных) | 55 |  |
| **Тема 4.** Тестирование информационной системы | 20 |  |
| **Тема 5.** Приемо-сдаточные мероприятия | 14 |  |

**Характеристика учебной/профессиональной деятельности обучающегося во время учебной практики (по профилю специальности):**

Общие и профессиональные компетенции, предусмотренные программой практики, **освоены /** не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Коннова А.Е. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Дата «27» апреля 2024 г.