Министерство науки и высшего образования РФ

Федеральное государственное образовательное учреждение

высшего образования

«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Кафедра АСУ

Лабораторная работа №1

по дисциплине

«Объектно-ориентированное моделирование

и программирование»

«Система контроля версий Git: совместная разработка ПО»

Выполнили:

студенты гр. ПИ-101Бзу

Уразбахтин Т.А.

Шаимов А.Р.

Проверил:

преподаватель

Казанцев А.В.

г. Уфа 2024

**Лабораторная работа №1**

**Система контроля версий Git: совместная разработка ПО**

Цель работы: ознакомление с системой контроля версий Git, а также с возможностями использования IDE Eclipse для работы с веб-сервисом GitHub.

Ход работы

Таблица 1 – Информация о команде

|  |  |
| --- | --- |
| Модератор | Разработчик |
| Шаимов Айдар Радикович  username - AidarShaimov | Уразбахтин Тимур Альбертович  username - owner08 |

Таблица – 1 Информация о команде

Таблица 2 – Зоны ответственности

|  |  |
| --- | --- |
| Зона ответственности модератора | Зона ответственности разработчика |
| Реализация главного класса приложения (main метод), включая графический интерфейс. Сборка проекта. | Реализация абстрактных классов и методов.  Реализация классов на основе абстрактных классов, переопределение абстрактных методов. |

Таблица 2 – Зоны ответственности

1. Изучили теоретическую часть.

2. Установили Eclipse, выбрав «Eclipse IDE for Java Developers» в соответствии с рисунком 1.

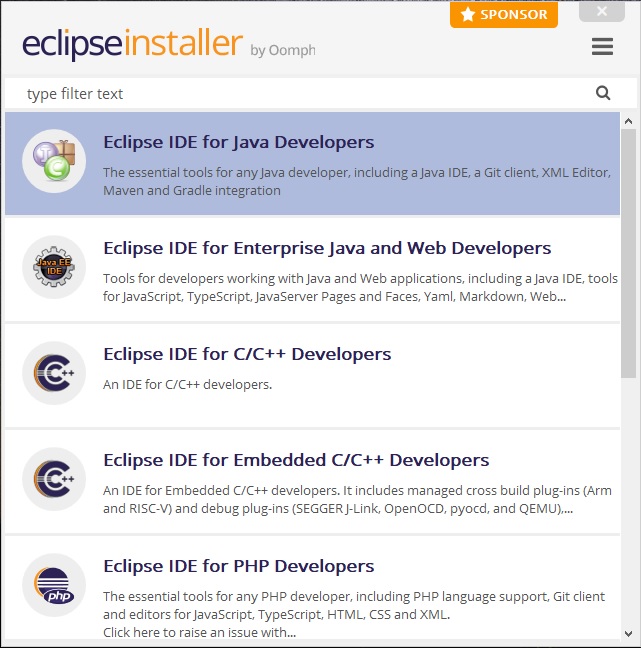


Рисунок 1 – Установка Eclipse

3. Запустили среду Eclipse в соответствии с рисунком 2.

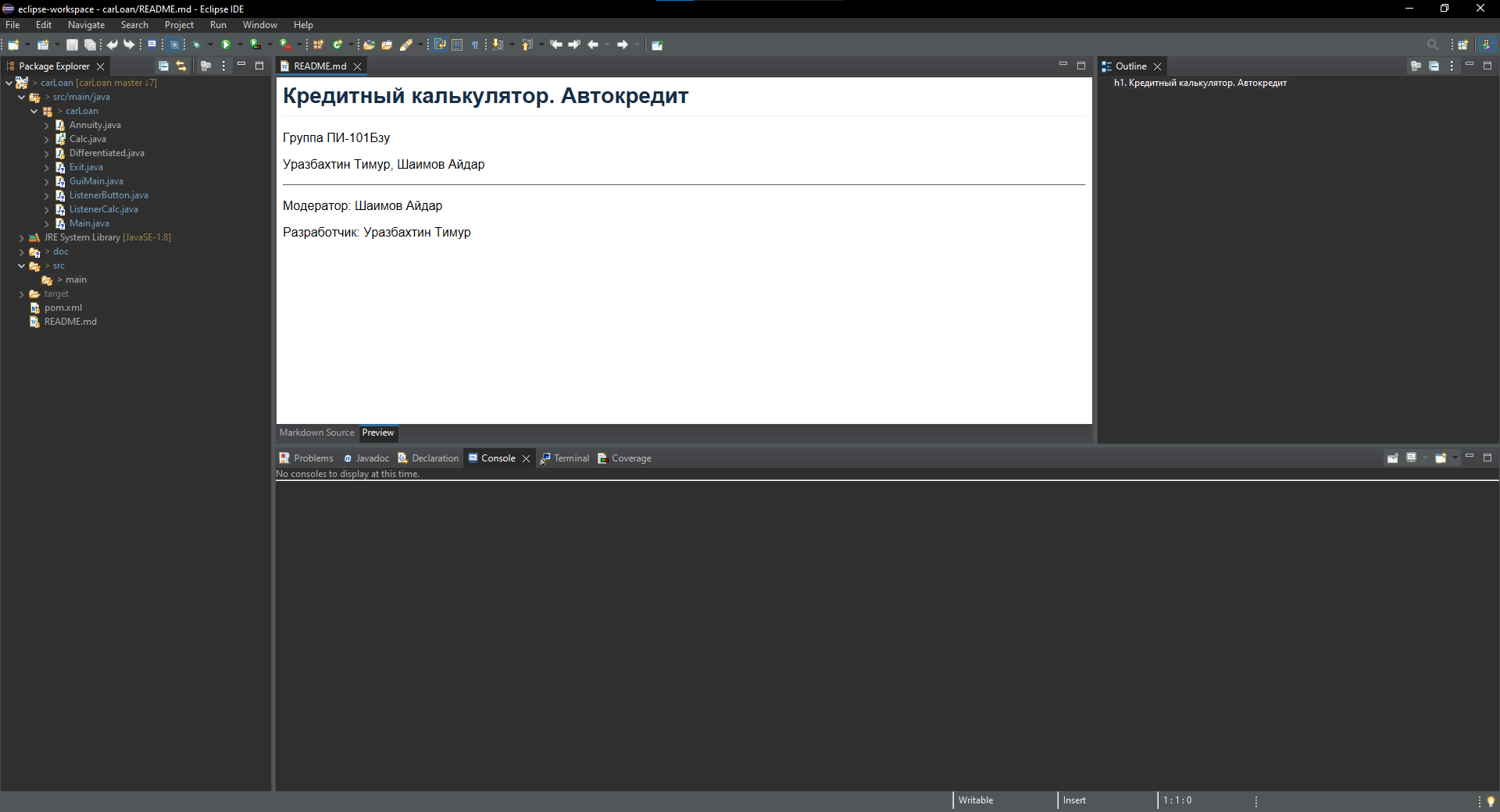


Рисунок 2 – Запуск Eclipse

4. Разделились по ролям в команде: модератор и разработчик. Модератор: Шаимов Айдар. Разработчик: Уразбахтин Тимур.

5. Модератор создал репозиторий и подключил к нему разработчика в соответствии с рисунком 3.

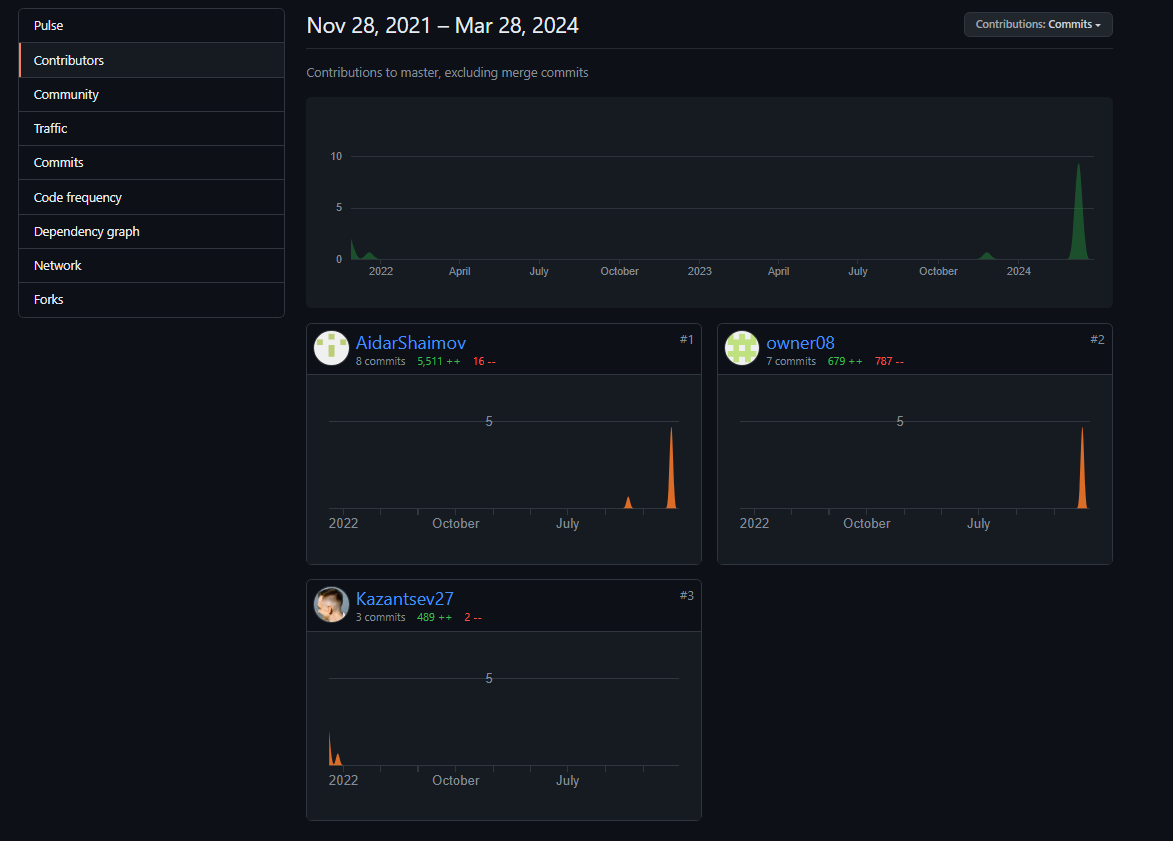


Рисунок 3 – Подключение в репозиторий

6. В соответствии с зоной ответственности разработали программное обеспечение. Создание каждого класса сопровождалось фиксацией изменений (commit) в репозитории и дальнейшей выгрузкой (push) на github в соответствии с рисунком 4.

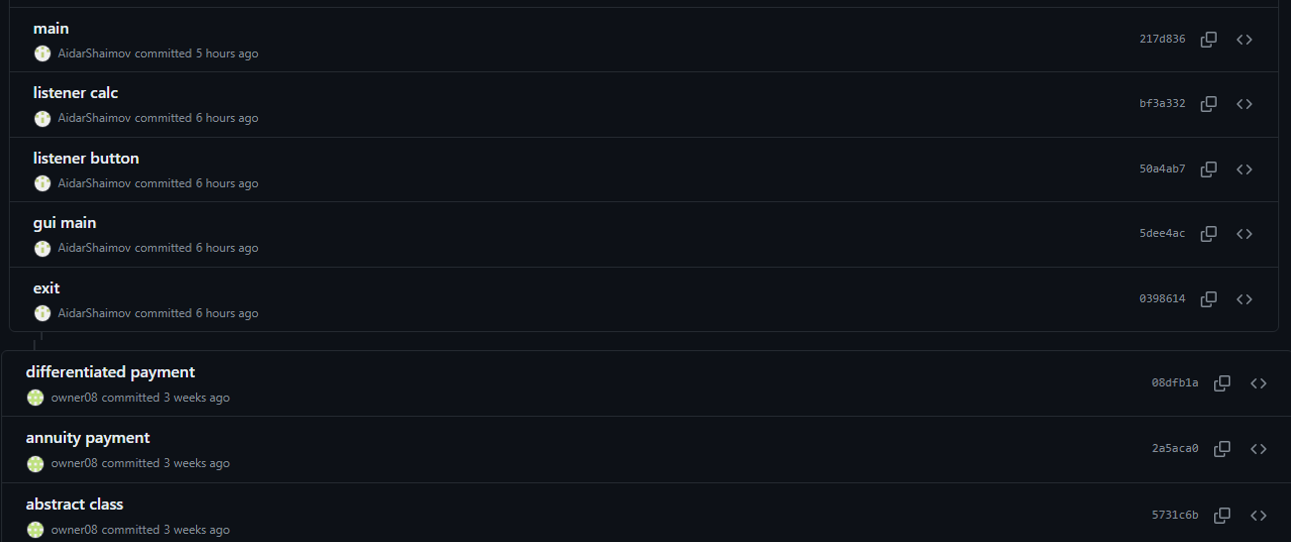


Рисунок 4 – Фиксация изменений

7. Собрали проект и поместили jar-файл в папку «App» в основной структуре проекта в соответствии с рисунком 5.

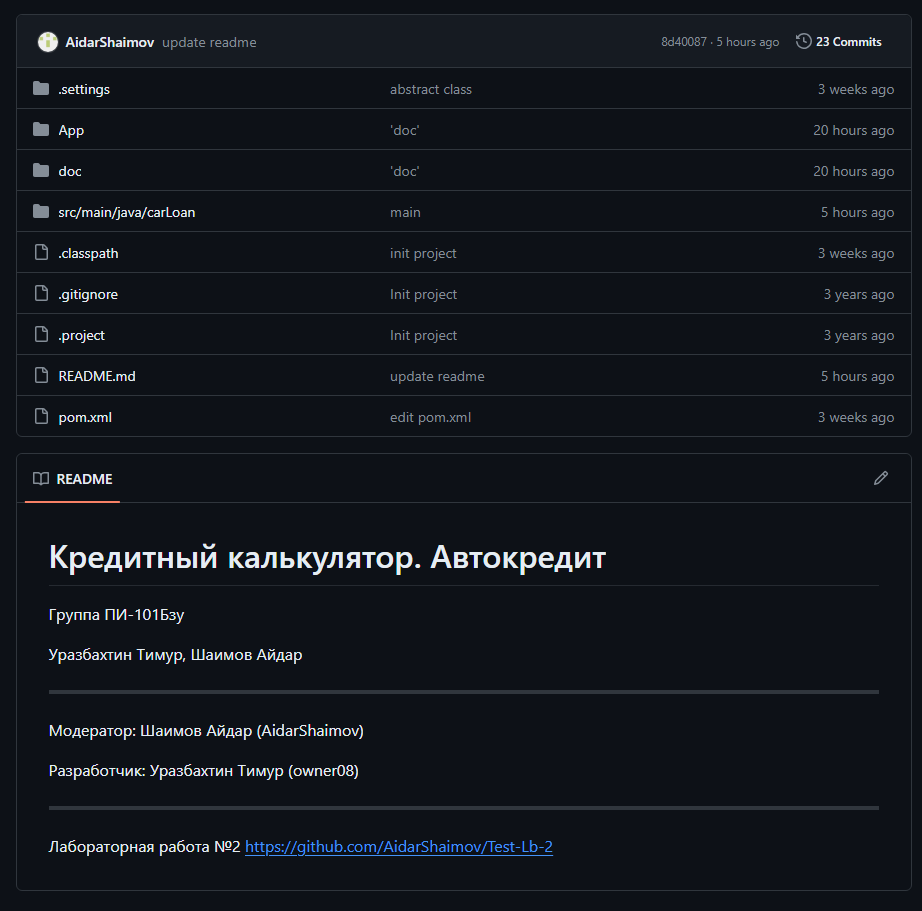


Рисунок 5 – Собранный проект

**Заключение**: ознакомились с системой контроля версий Git, а также с возможностями использования IDE Eclipse для работы с веб-сервисом GitHub.

**Контрольные вопросы.**

**1. Что такое Git?**

Git - это распределенная система контроля версий, которая позволяет отслеживать изменения в файлах и управлять этими изменениями.

**2. Как добавить поддержку системы контроля версий Git в Eclipse?**

Чтобы добавить поддержку системы контроля версий Git в Eclipse, нужно открыть Eclipse и перейти в «Справка» -> «Install New Software…» В открывшемся окне нажать «Add…», ввести имя нового соединения и нажать «ОК». Затем в поле «Work with» выбрать добавленное соединение и нажать «Enter». В списке доступных пакетов найти и установить пакеты «EGit», «Egit Feature Update Site» и «Subclipse» (если нужно использовать Subversion). Нажать «Next», принять условия лицензионного соглашения и нажать «Finish» для установки пакетов. После установки перезапустить Eclipse.

**3. Как получить доступ к репозиторию на Github?**

Создать учетную запись на GitHub и клонировать репозиторий на компьютер, выполнив следующие действия:

1) Создать учетную запись на GitHub или войди в уже существующую.

2) Перейти на страницу нужного репозитория и нажать кнопку “Clone” (клонировать) или “Download” (скачать).

3) Выбрать способ клонирования (SSH или HTTPS) и скопировать URL-адрес репозитория.

4) Открыть терминал или командную строку на компьютере.

5) Ввести команду “git clone” с аргументом в виде скопированного URL-адреса и нажать Enter.

6) После выполнения команды репозиторий будет загружен и создан его локальная копия.

**4. Как изменить комментарий?**

1) Сначала нужно спрятать изменения с помощью команды git stash.

2) Затем можно просмотреть список коммитов с помощью git log --oneline.

3) Для изменения коммита нужно выполнить команду git rebase -i и указать нужный коммит.

4) В текстовом редакторе нужно изменить слово pick на edit у коммита, который нужно изменить.

5) После сохранения изменений можно снова вернуться в командную строку или отредактировать файлы.

6) Отредактированные файлы нужно добавить в индекс с помощью команды git add.

7) Для изменения нескольких коммитов нужно изменить слово pick на edit у нескольких коммитов в списке.

**5. Как зафиксировать изменения в Eclipse?**

1) В Eclipse открыть проект, в котором надо зафиксировать изменения.

2) Перейти в Package Explorer (Alt+Shift+Enter, если Package Explorer не открыт).

3) Найти папку с файлами, в которых надо зафиксировать изменения (обычно это папка src или main).

4) Выбрать файлы, которые надо зафиксировать (Ctrl и выбрать файлы).

5) Щелкнуть правой кнопкой мыши на выбранных файлах и выбрать “Team” -> “Commit…” из контекстного меню.

6) В открывшемся окне ввести комментарий к фиксации изменений и нажать “OK”.

7) Изменения будут зафиксированы и добавлены в систему контроля версий.