**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа энергетики

Отделение электроэнергетики и электротехники

Направление: 09.04.03 Прикладная информатика

Дисциплина: Программирование и основы алгоритмизации

Отчет

по лабораторной работе №9

**Знакомство с системой управления версиями**

Вариант 12

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. О-5КМ41 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Рустамов Р.А. |
| Проверил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Гончаров А.С. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск-2025

# ХОД РАБОТЫ

Для подключения GitHub-аккаунта к Visual Studio в верхнем меню выбирается пункт *Файл → Параметры учётной записи…*.

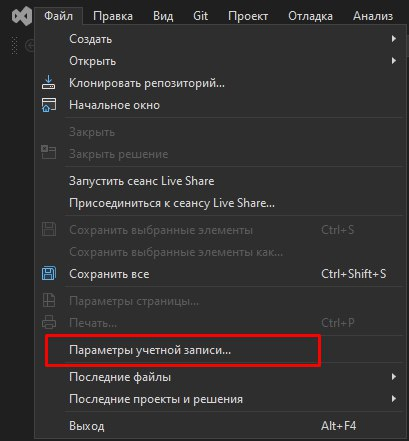


Рисунок –Способ подключения GitHub через "параметры учетной записи"

В открывшемся окне показано, что к Visual Studio можно привязать сразу несколько сервисов. Здесь активна синхронизация с *Microsoft Account* и выполнен вход в GitHub.

|  |  |
| --- | --- |
|  |  |

Рисунок – Способ подключения GitHub через вход в учетную запись

После выбора GitHub выполняется переход в браузер, где нужно подтвердить вход в свою учётную запись GitHub.

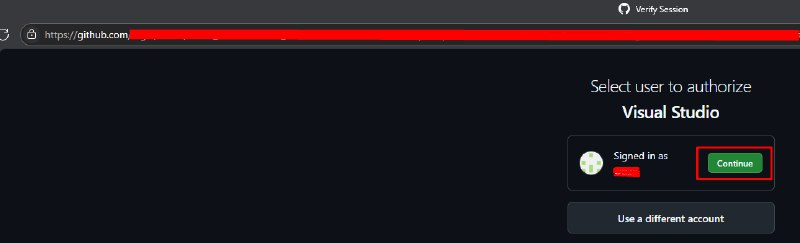


Рисунок –Переход в браузер для подтверждения входа

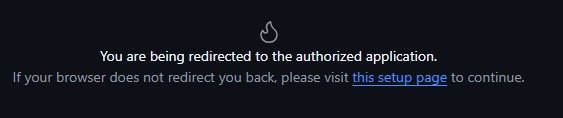


Рисунок – Информация об успешном входе

Пользователь перенаправляется обратно в Visual Studio после успешного входа в GitHub. На этом шаге выполняется настройка нового репозитория. Указывается:

• локальный путь до проекта,

• имя репозитория (Лабы по C#),

• описание,

• видимость (в примере — Private).

После этого проект можно сразу отправить в GitHub.

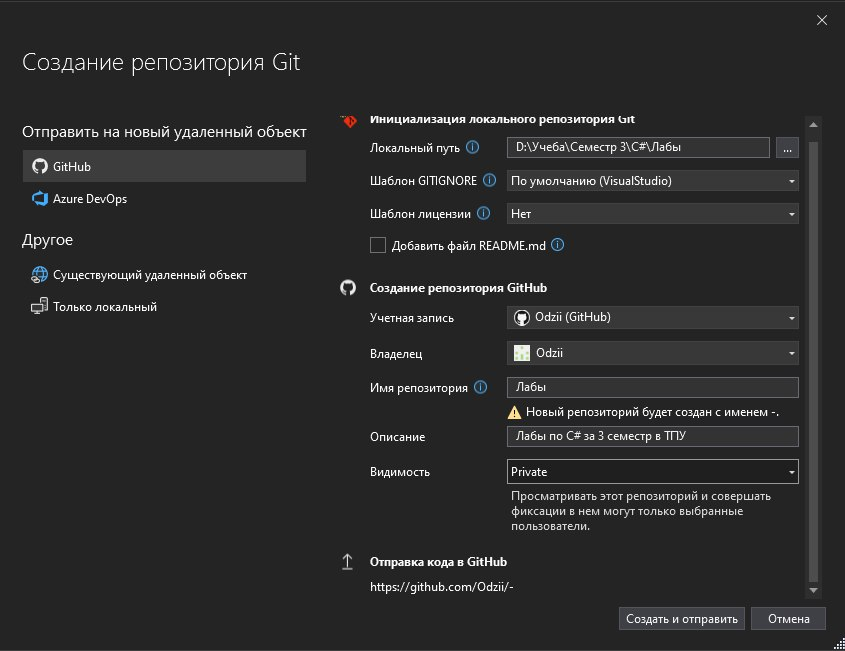


Рисунок –Создания репозитория в VS 2022 И Github

Завершающий шаг — проверка созданного репозитория в браузере. В нём отображаются загруженные папки и файлы лабораторных работ, ветки, список коммитов и служебные файлы (.gitignore, .gitattributes).

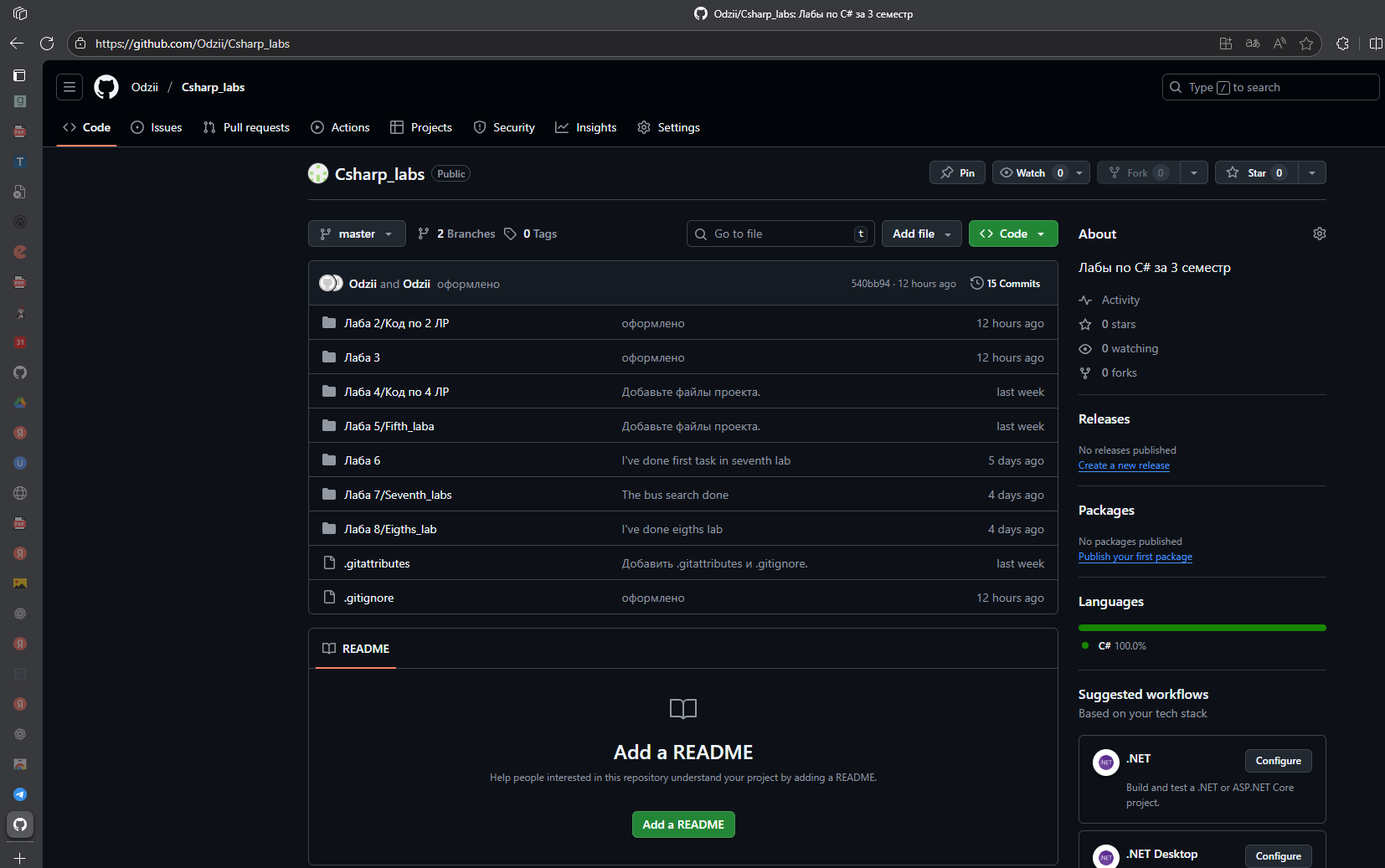


Рисунок – Результат фиксации изменения и отправки данных на Git

В случае, если репозиторий сначала был создан на Github, то необходимо скопировать ссылку и выполнить клонирование. На сайте GitHub для клонирования проекта необходимо скопировать URL репозитория (в примере — https://github.com/Odzii/Csharp\_labs.git).

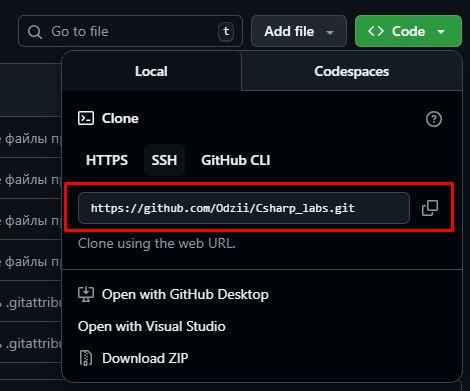


Рисунок – Получение ссылки на репозиторий GitHub

В Visual Studio в верхнем меню выбирается пункт *Git → Клонировать репозиторий…*.

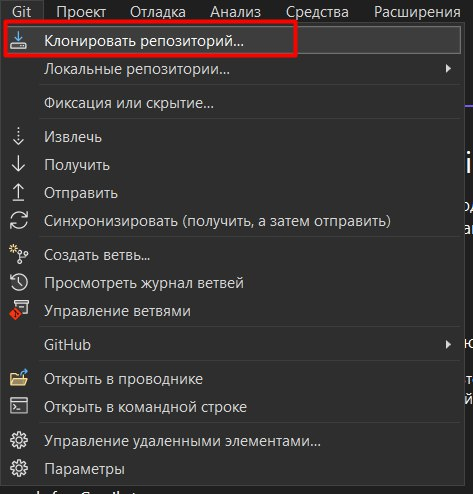


Рисунок – Выбор команды клонирования репозитория в Visual Studio

В открывшемся окне вводится скопированный URL-адрес и выбирается папка на локальном компьютере, куда будет загружен репозиторий.

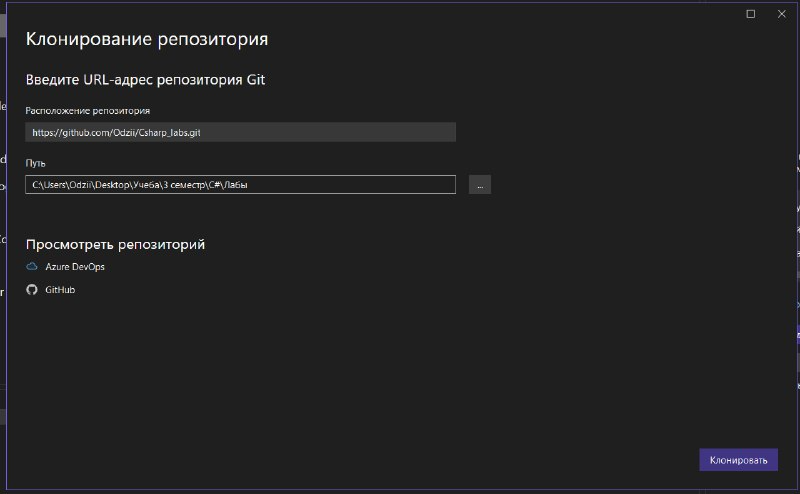


Рисунок – Ввод URL и указание пути для клонирования

После подтверждения выполняется загрузка файлов с GitHub. В результате локальная копия репозитория создаётся на диске и сразу открывается в Visual Studio.

|  |  |
| --- | --- |
|  |  |

Рисунок – Рисунок 10 –Завершение клонирования

На последнем скриншоте видно, что проект успешно открыт в Visual Studio. В *Обозревателе решений* отображаются файлы, которые были загружены из удалённого репозитория. Теперь разработчик может редактировать, запускать и синхронизировать проект с GitHub.

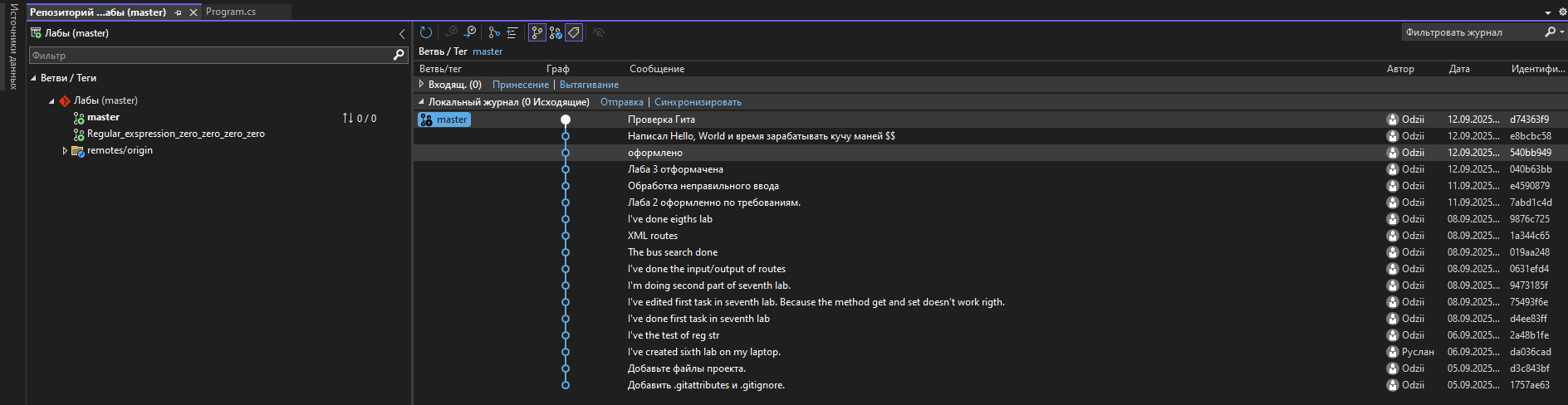


Рисунок – Открытый репозиторий в Visual Studio

«На рисунке показан результат выгрузки файлов проекта в удалённый репозиторий GitHub. В репозитории *Csharp\_labs* размещены папки с лабораторными работами и служебные файлы .gitignore, .gitattributes. В истории отображаются зафиксированные изменения (коммиты), что подтверждает корректную работу синхронизации между Visual Studio и GitHub. Таким образом, была проверена успешность подключения среды разработки Visual Studio к системе контроля версий Git.»

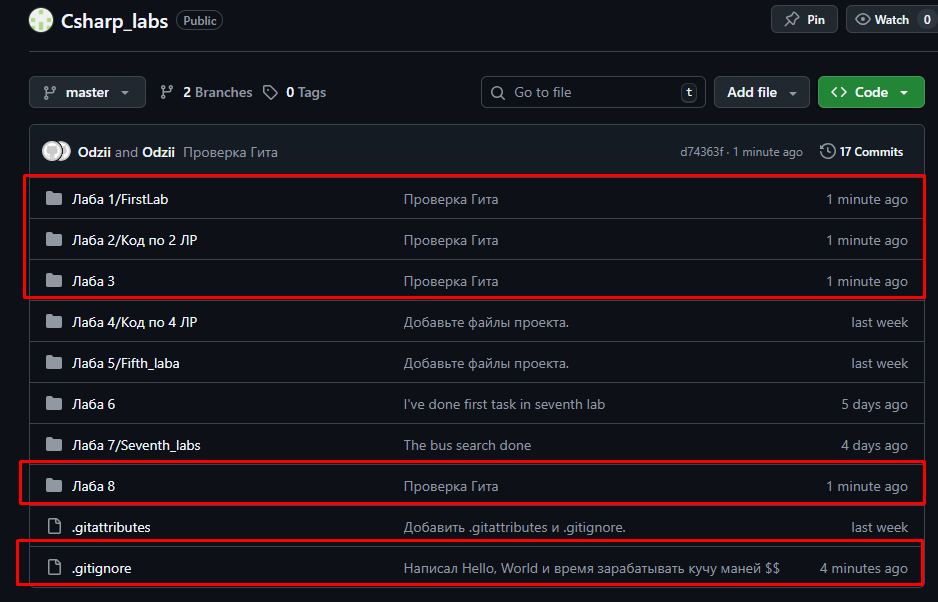


Рисунок –Результат отправки изменений

В интерфейсе GitHub выбран раздел *Issues*, который используется для постановки задач, фиксации ошибок и ведения обсуждений по проекту.

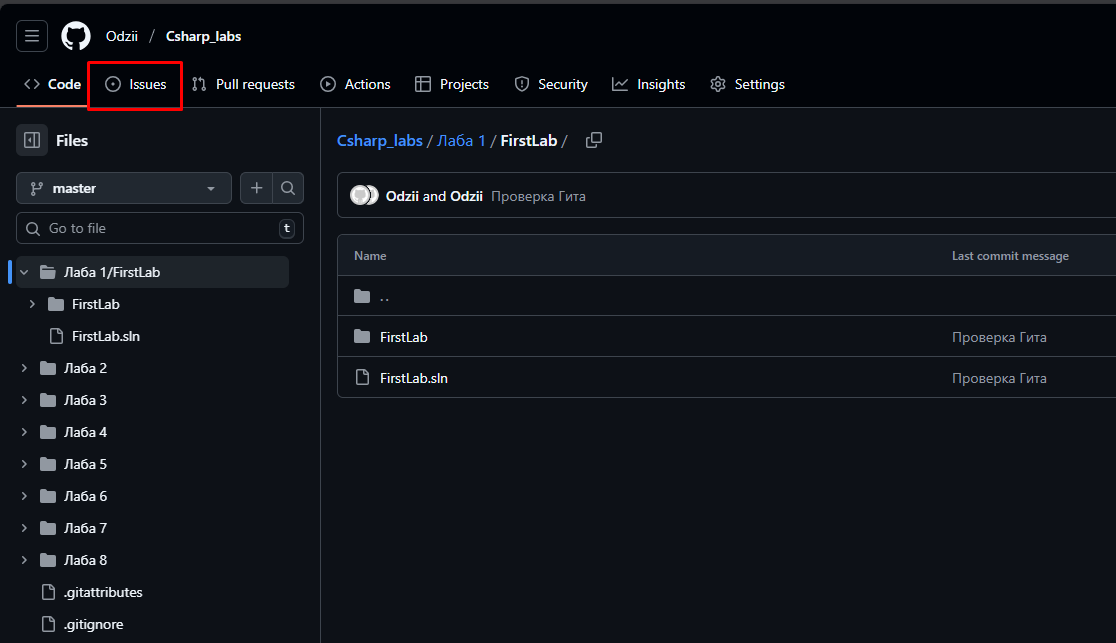


Рисунок – Переход к разделу Issues

В разделе *Issues* нажата кнопка *New issue*. Это позволяет задать новую задачу для проекта.

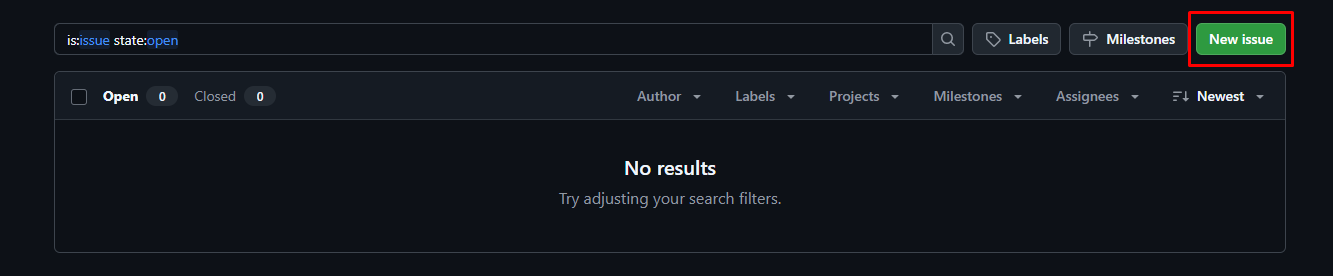


Рисунок – Создание новой задачи

Создана задача с названием *«Create rus ver Hello World»* и описанием *«Создать русскую версию Hello World»*. Статус задачи – *Open*, что означает её актуальность.

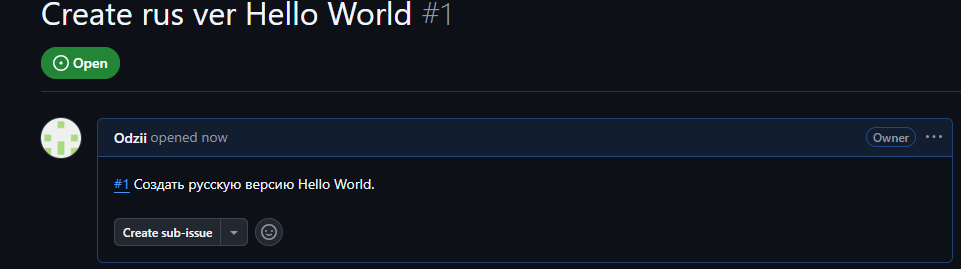


Рисунок – Пример созданного Issue

В Visual Studio при выполнении фиксации изменений (commit) в комментарии добавлено closes #1. Это связывает коммит с задачей под номером 1 на GitHub.

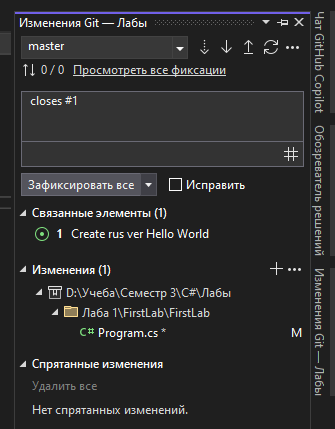


Рисунок –Привязка Issue к коммиту в Visual Studio

После отправки коммита и синхронизации проект с GitHub, задача автоматически закрывается. В истории репозитория отображается отметка *closes #1*, что подтверждает выполнение задачи.

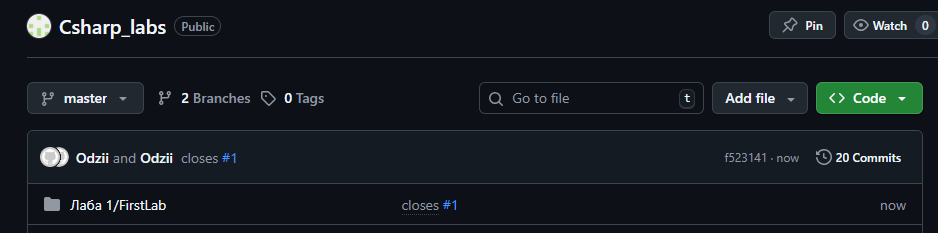


Рисунок –Закрытие Issue после фиксации

В меню *Git* выбрана команда *Создать ветвь…*. Эта функция позволяет создать отдельную ветку для разработки без изменения основной (master).

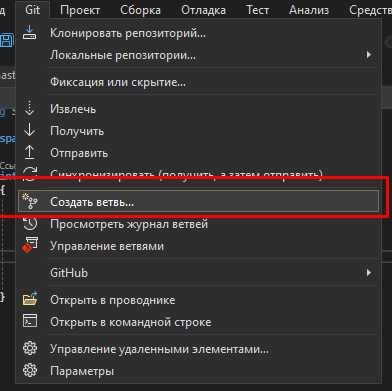


Рисунок – Выбор команды создания ветви в Visual Studio

В появившемся окне указано имя новой ветви (*GitTest*) и выбрана её основа — *master*. Опция *Извлечь ветвь* позволяет сразу переключиться на неё.

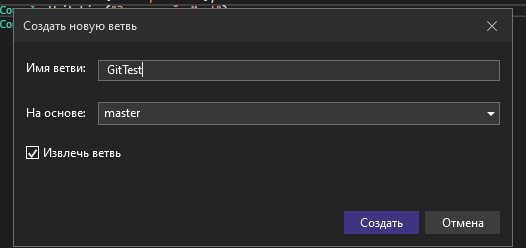


Рисунок – Создание новой ветви

После синхронизации с GitHub в разделе *Branches* отображаются все ветки репозитория:

* *master* — основная ветвь;
* *GitTest* — только что созданная ветка;
* дополнительные ветви, созданные ранее.

Отмечено, что ветка *GitTest* активна, а также указано время последнего обновления и статус синхронизации. В Visual Studio можно переключаться между ветками, просматривать их историю и выполнять слияния (merge). Это позволяет организовать параллельную разработку и экспериментировать с кодом без риска повредить основную версию проекта.

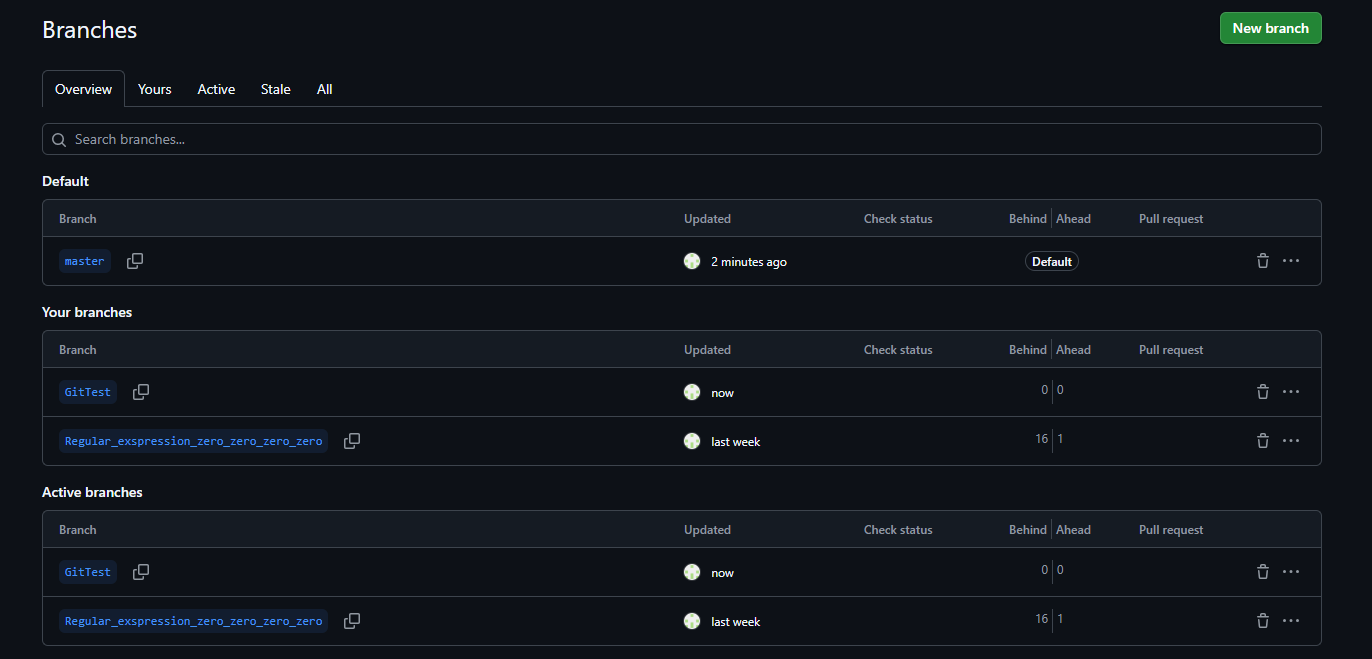


Рисунок – Активная ветка в GitHub

В *Обозревателе ветвей* Visual Studio выбрана ветка *GitTest* и команда *Слияние с Current Branch*.

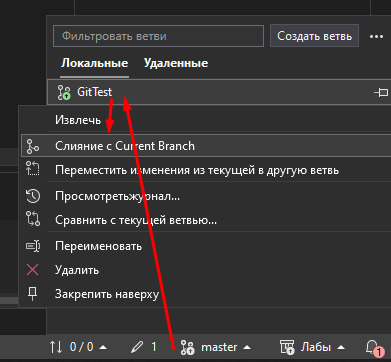


Рисунок – Вызов команды слияния ветвей

В диалоговом окне указано, что ветка *GitTest* будет объединена с основной веткой *master*. Отмечено, что слияние приведёт к изменению одного файла.

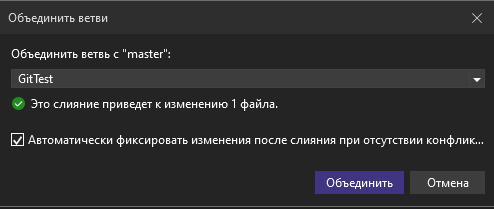


Рисунок – Окно подтверждения слияния

При слиянии обнаружены изменения в одном и том же файле (Program.cs). Visual Studio открыла окно сравнения:

* слева – изменения из ветки *GitTest*;
* справа – текущая версия в *master*;
* снизу – результирующий вариант, куда можно выбрать необходимые строки (входящие, текущие или объединённые).

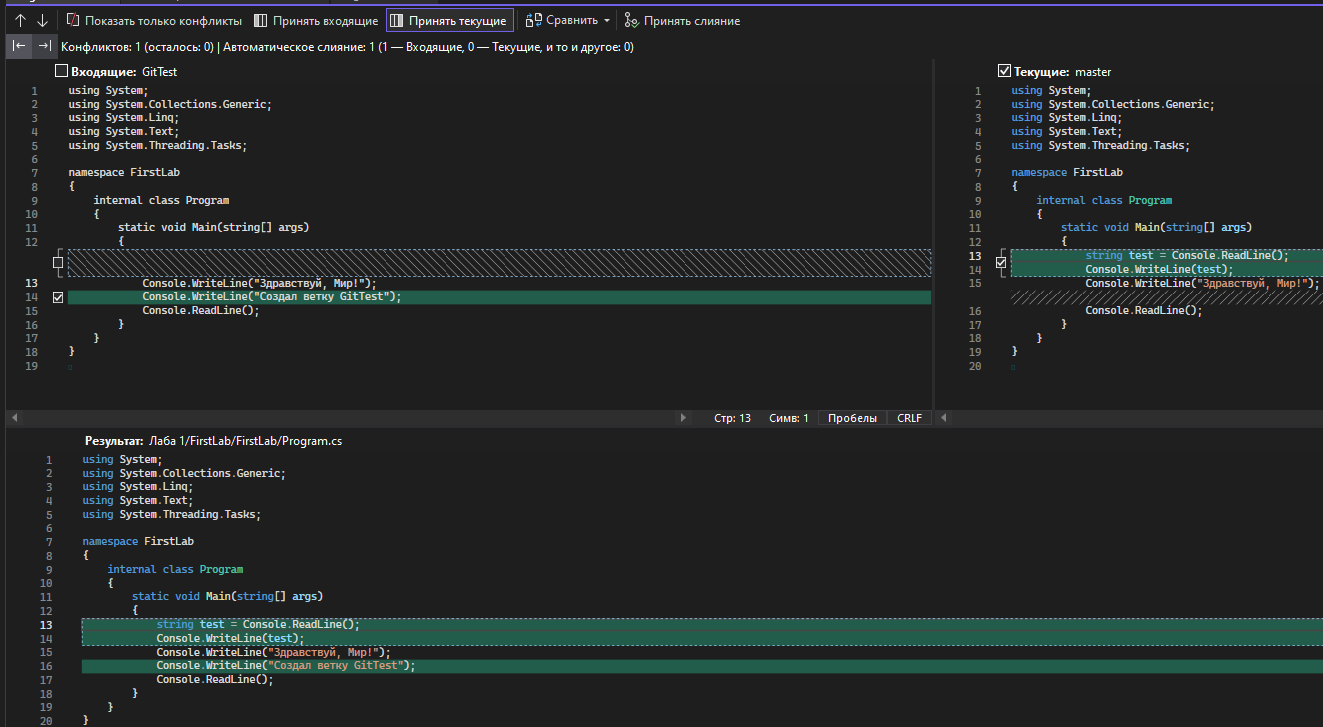


Рисунок – Разрешение конфликтов

В области управления Git Visual Studio вывела сообщение о наличии конфликта: *«слияние завершено с конфликтами»*. Пользователь должен вручную разрешить конфликт и зафиксировать изменения. В локальном журнале видно, что после разрешения конфликта и фиксации изменений коммит был добавлен в ветку *master*. В истории отобразились коммиты слияния (*merge commit*), что подтверждает успешное объединение веток.

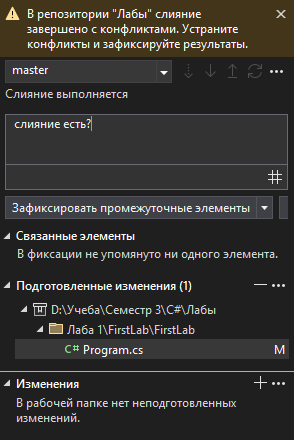


Рисунок – Оповещение о конфликте

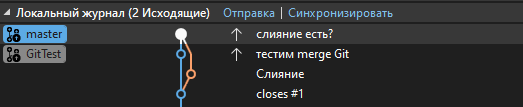


Рисунок – Синхронизация с GitHub

После завершения слияния и локальной фиксации изменения были отправлены на GitHub. Теперь в удалённом репозитории ветки объединены, а история изменений сохраняет все шаги разработки.

# ВЫВОД

В ходе выполнения лабораторной работы было изучено использование системы контроля версий Git и её интеграция со средой разработки Visual Studio 2022. Была выполнена авторизация в GitHub, создан и выгружен новый репозиторий, а также осуществлено его клонирование на локальный компьютер.В процессе работы были рассмотрены основные возможности взаимодействия Visual Studio с GitHub: фиксация изменений (commit), синхронизация локального и удалённого репозиториев, работа с задачами (*Issues*), создание и использование веток, а также их слияние (*merge*), включая разрешение конфликтов.