МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе №1**

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ *Ассистент кафедры ЗИ*

*Бурлаков И.Е. Исаев Г. А.*

«\_\_» \_\_\_\_\_\_ 2024г«\_\_\_» \_\_\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2024

**Цели и задачи работы**: изучение платформы GitHub и практическое

использование Git для хранения, обновления и распространения исходного

кода проекта.**Задание №1 к работе**: Разработать алгоритм решения задачи по индивидуальному заданию.

Геометрические фигуры. Реализовать функции вычисления параметров для следующего геометрической фигур:

b. Треугольник – периметр, площадь по формуле Герона, проверка на равнобедренность;

**Методика выполнения работы**:

1. Разработать алгоритм решения задачи по индивидуальному заданию.
2. Написать и отладить программу решения задачи.
3. Протестировать работу программы на различных исходных данных.
4. Продемонстрировать работу с Git и GitHub на примере написанной программы.
5. По запросу преподавателя быть готовым модифицировать/добавить функционал программы.
6. Ответить на теоретические вопросы к лабораторной работе на выбор преподавателя.

**Листинг программы**:

https://github.com/JamesSevil/laba1.git

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

/// Периметр треугольника

cout << "Введите стороны треугольника:\n";

cout << "Сторона А = ";

int sideA = 0;

cin >> sideA;

cout << "Сторона B = ";

int sideB = 0;

cin >> sideB;

cout << "Сторона C = ";

int sideC = 0;

cin >> sideC;

cout << "Периметр треугольника = " << sideA + sideB + sideC;

/// площадь по формуле Герона

int Ploshad = sideA + sideB + sideC;

float T = (sideA + sideB + sideC) / 2;

cout << "\nПлощадь по формуле Герона = " << (sqrt(T \* (T - sideA) \* (T - sideB) \* (T - sideC)));

/// Равнобедренный треугольник ли

if (sideA == sideC || sideB == sideC || sideA == sideB)

{

cout << "\nТреуголник равнобедренный";

}

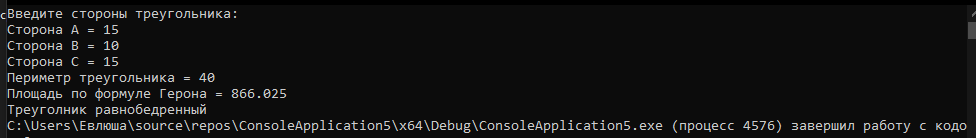
else {

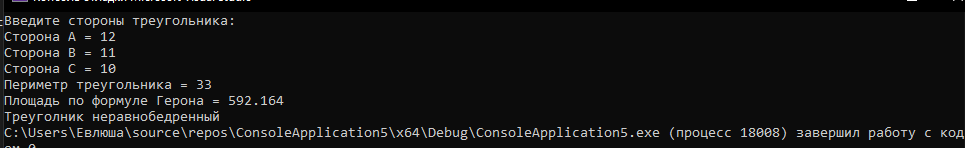
cout << "\nТреуголник неравнобедренный ";

}

}

**Результат работы программы:**

****

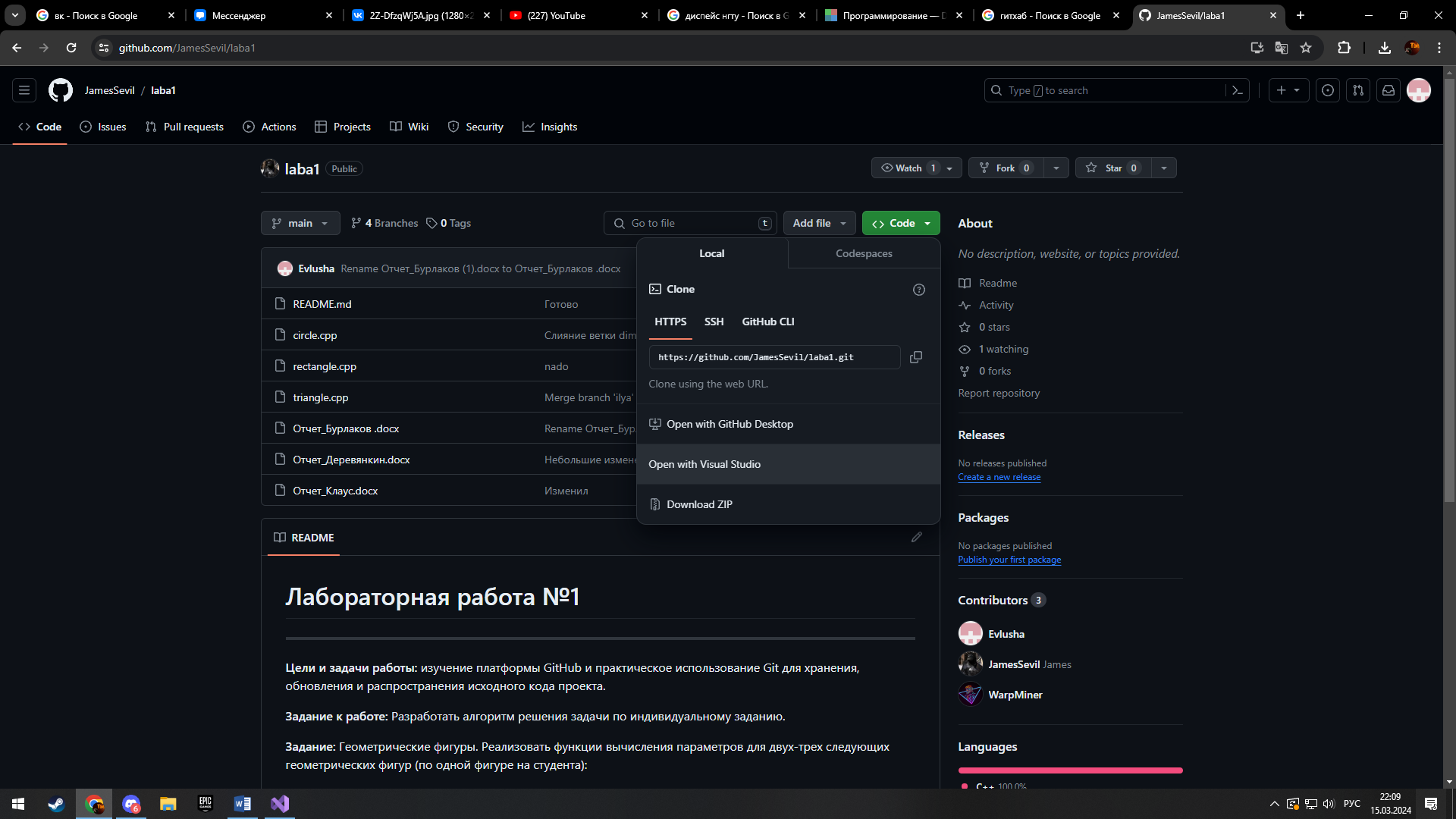
****

**Полное описание всех этапов процесса работы:**

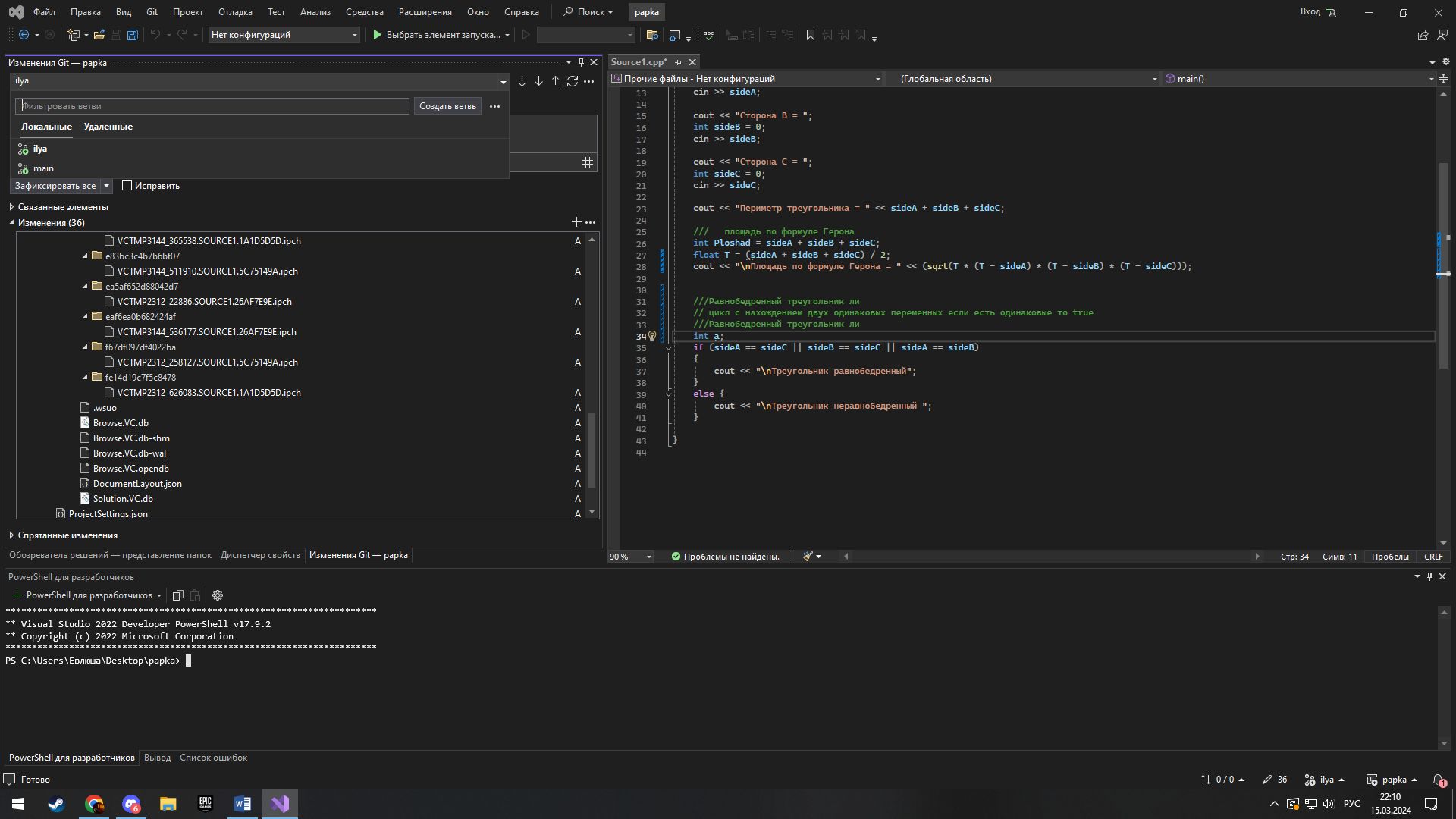
1. Создание пустого репозитория в GitHub:
2. Клонирование репозитория в рабочую папку, открытие проекта в текстовом редакторе (Visual Studio):
3. Создание отдельной ветки “ilya” и файла triangle.cpp, в котором будет оформляться код. Создаём отдельную ветку, чтобы избежать помех в работе в команде:
4. Оформление кода, доработка кода, сохранение и отладка.
5. Загрузка отдельной ветки “ilya” с файлом triangle.cpp в репозиторий. Загрузка осуществлялась через терминал текстового редактора:
6. Создание файла [README.md](https://vk.com/away.php?to=http%3A%2F%2FREADME.md&cc_key=) в основной ветке “main”. Оформление файла, сохранение и загрузка в репозиторий:
7. Выполнение слияния дополнительной ветки “ilya” в основную “main”. Сохранение и загрузка в репозиторий;

**Описание использованных функций :**

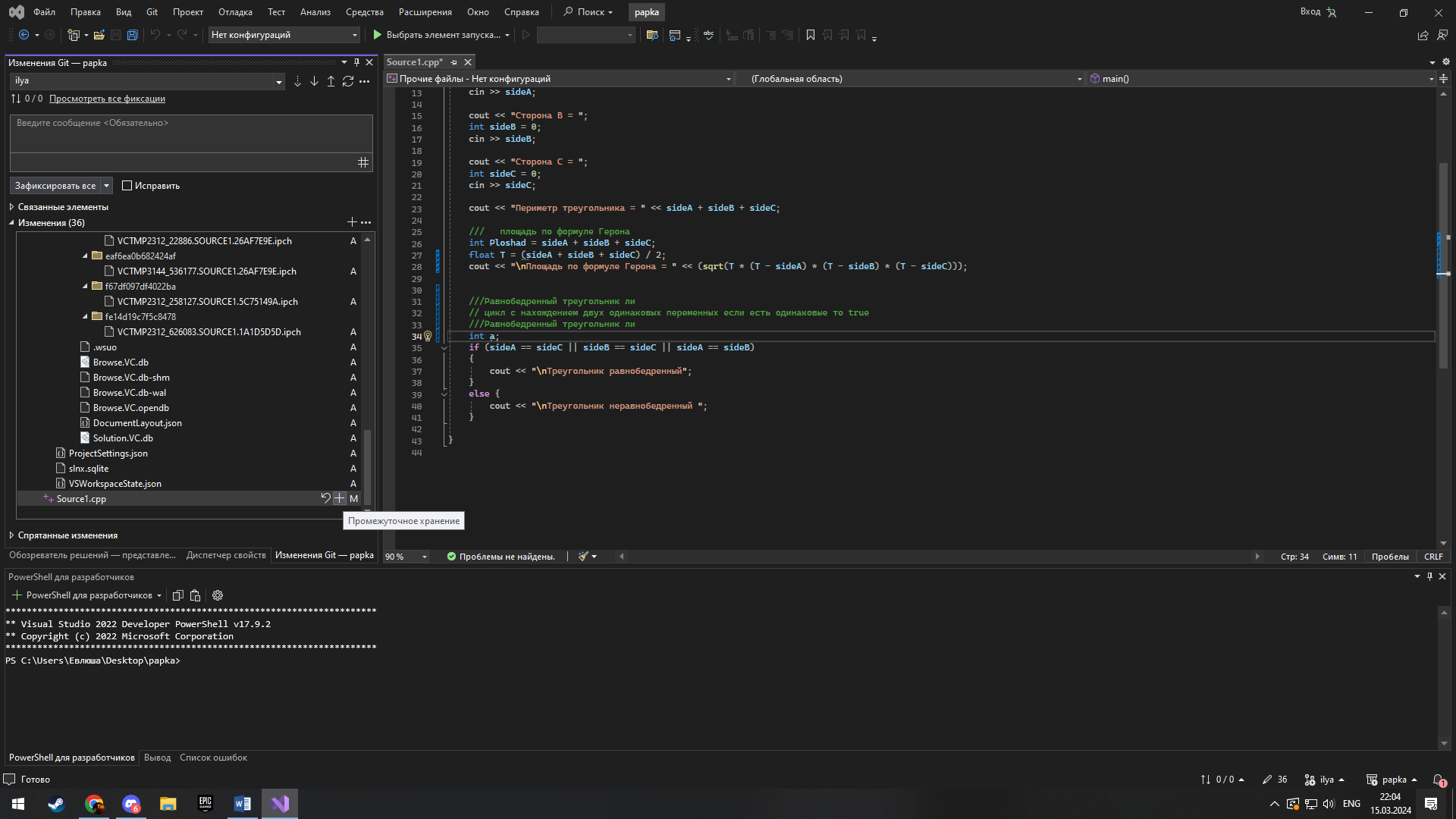
1.Клонирования репозитория



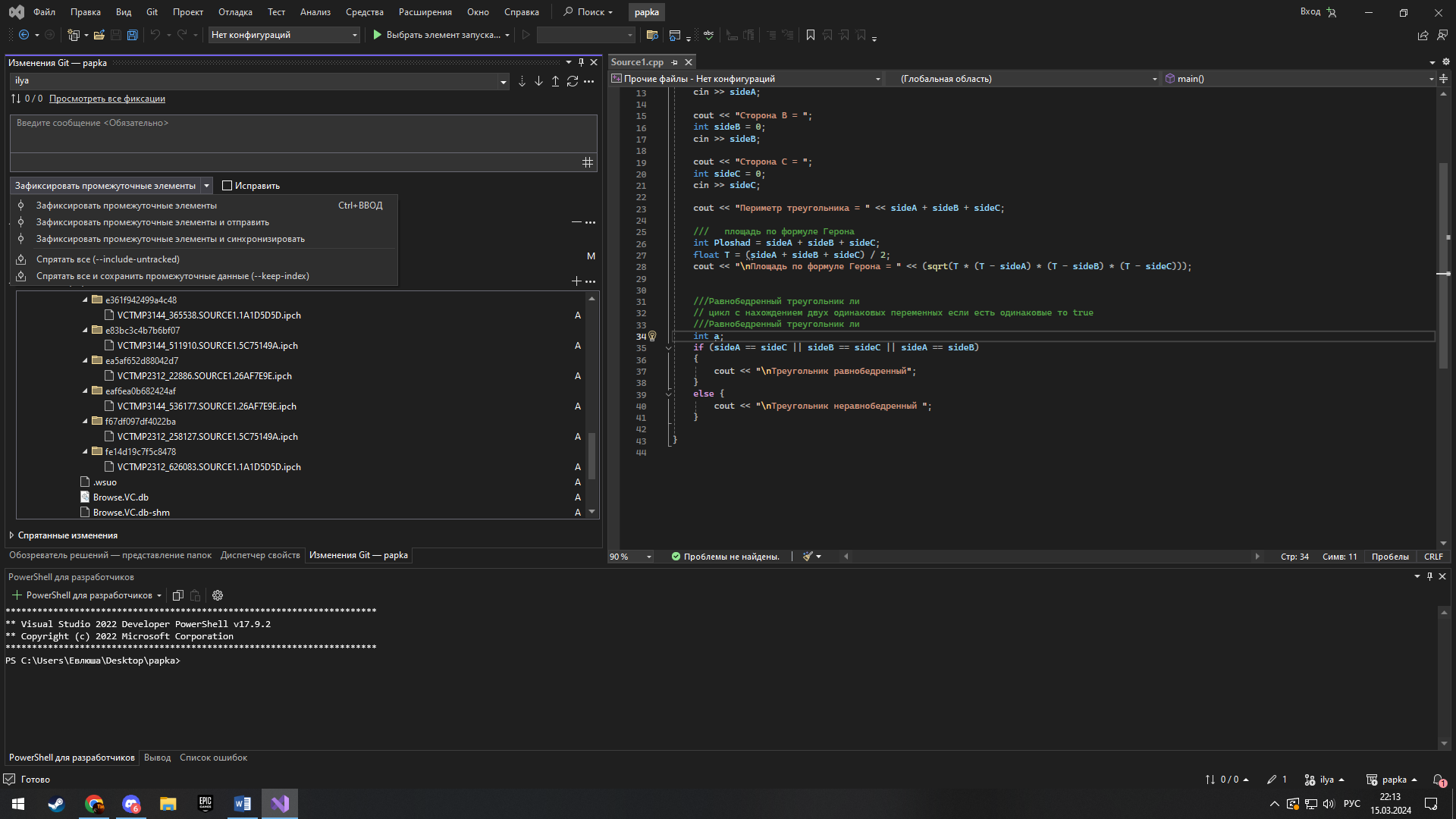
2.Создание ветви



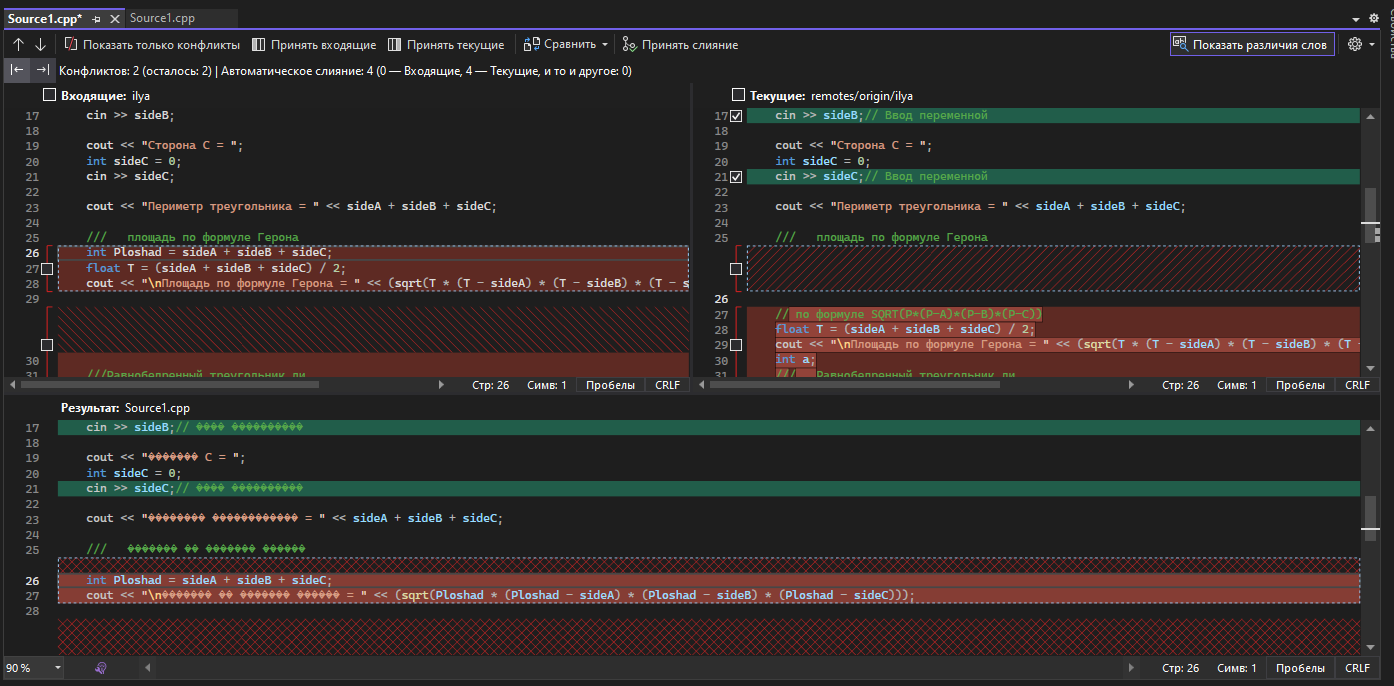
3.После изменения файла нужно сделать промежуточное хранение

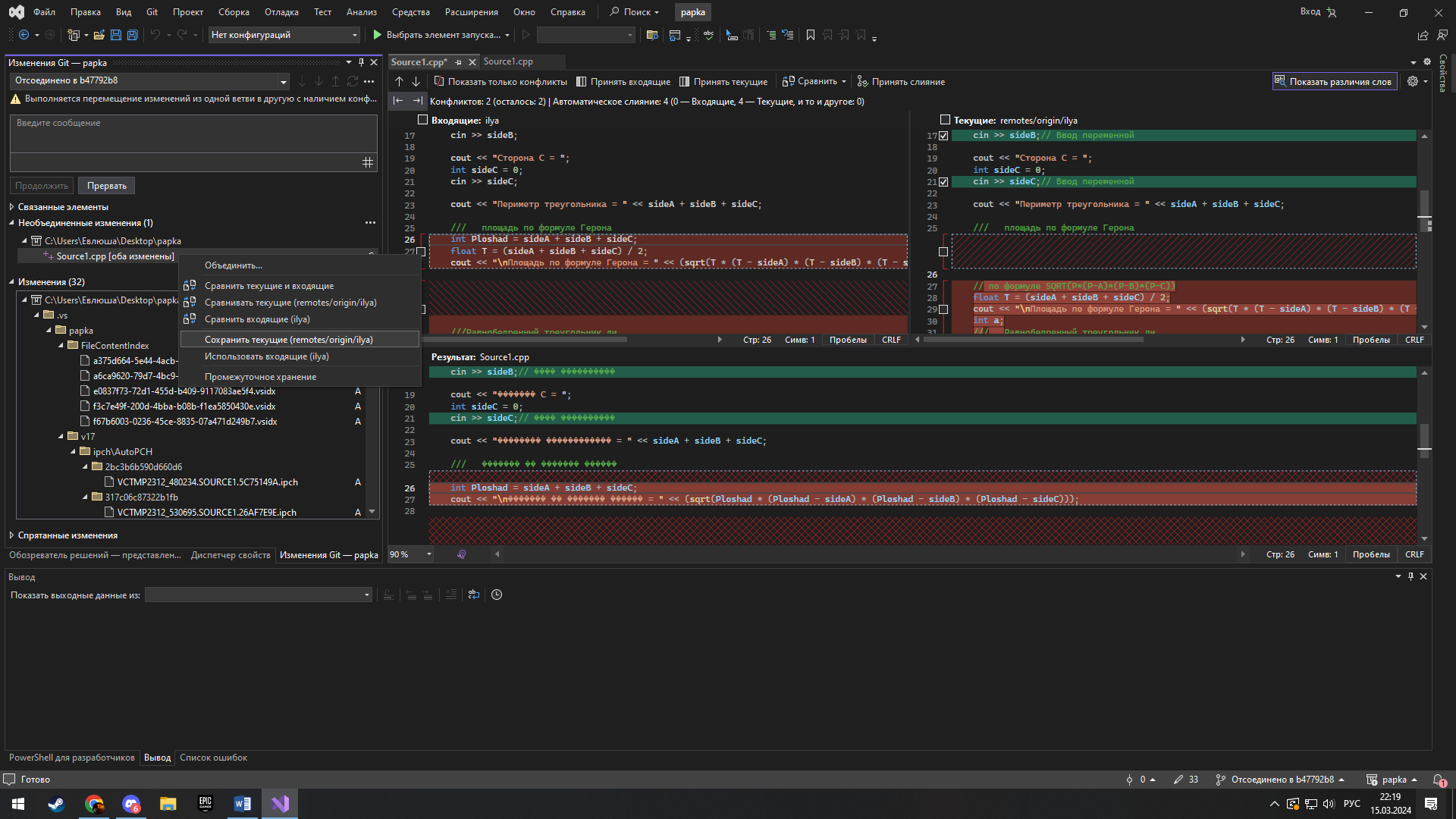


4.После промежуточного сохранения можно сразу отправить на ветку

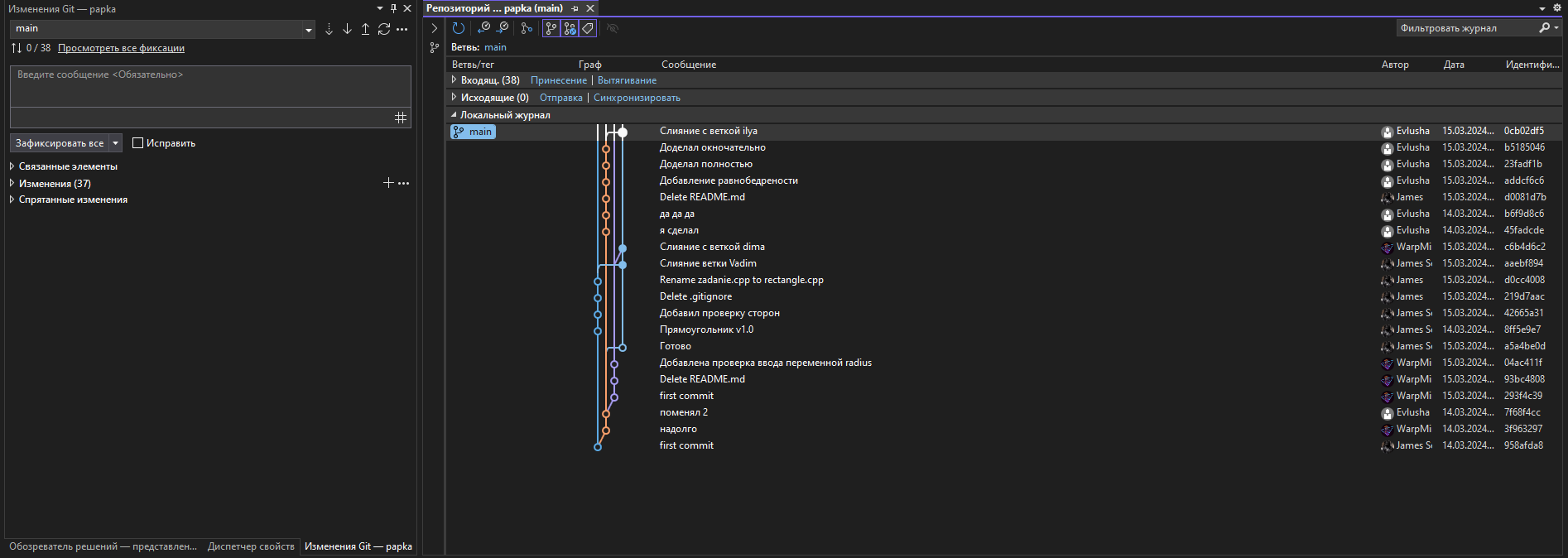


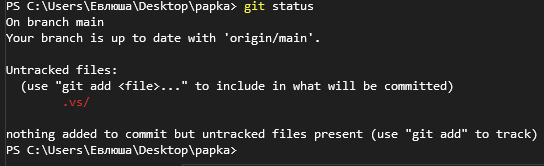
5.Решени конфликтов при появление такоого сообщения нужно вручную подтвердить решение конфликта



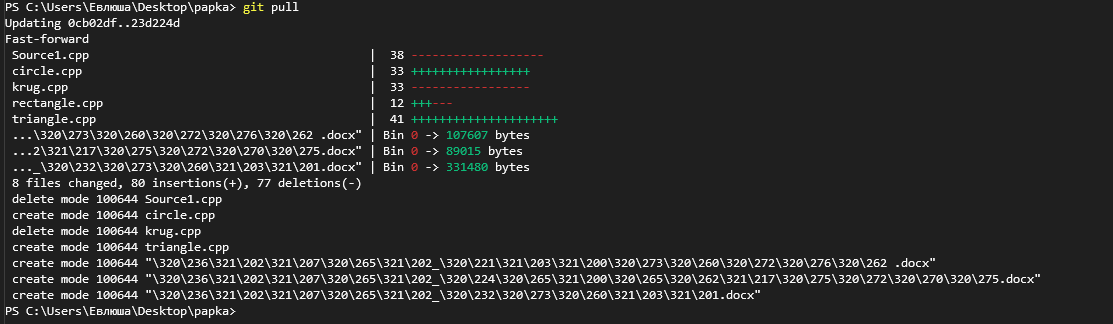


5. Таким же методом делается слияние веток

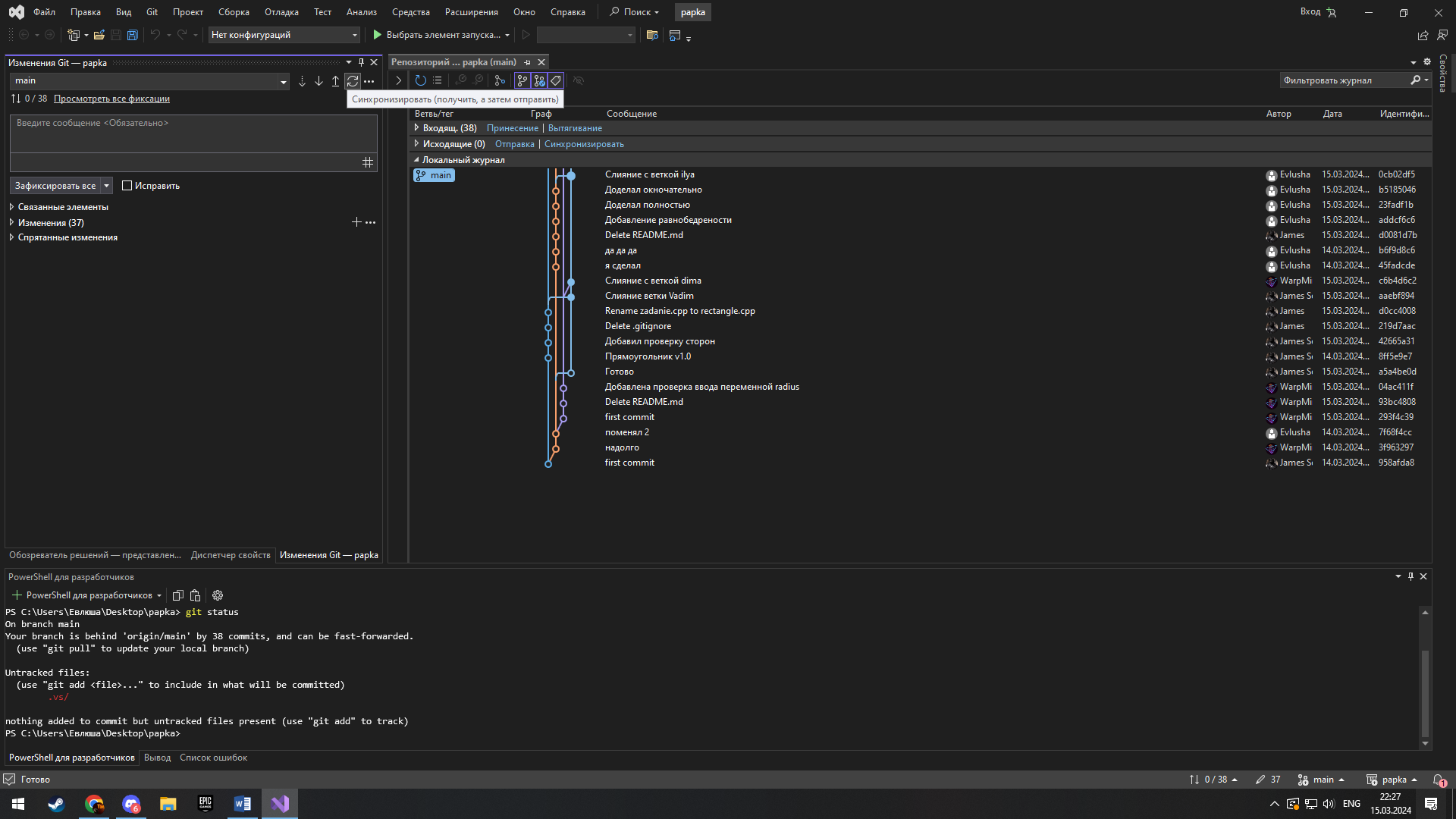


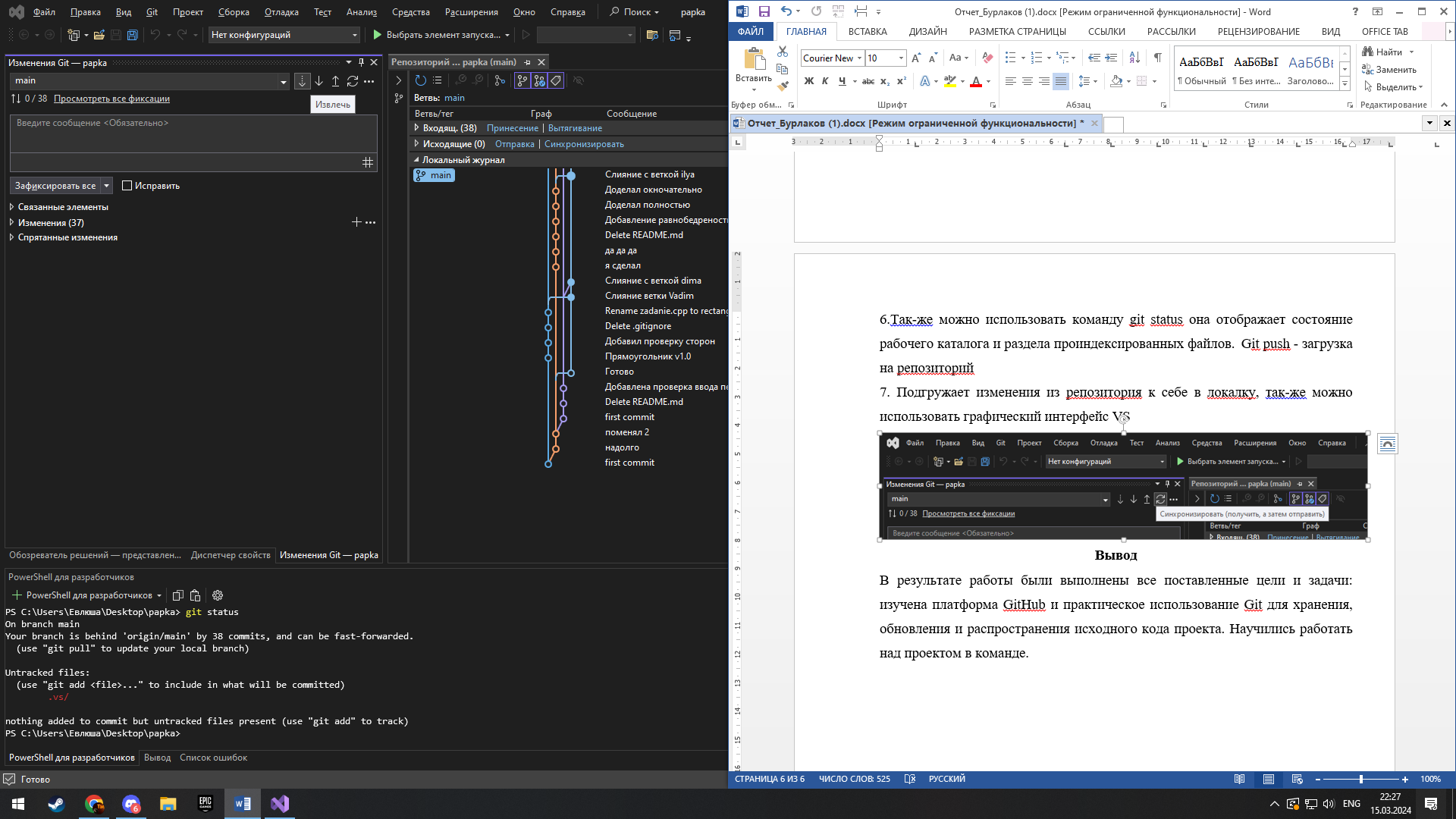
6.Так-же можно использовать команду git status она отображает состояние рабочего каталога и раздела проиндексированных файлов. 

7.Git push - загрузка на репозиторий



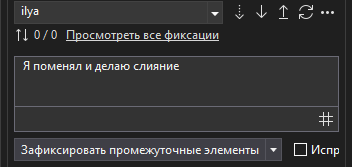
8. Подгружает изменения из репозитория к себе в локалку, так-же можно использовать графический интерфейс VS



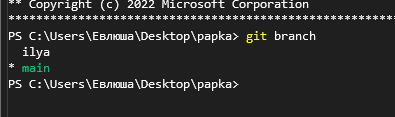


9. git merge - осуществляет слияние веток.

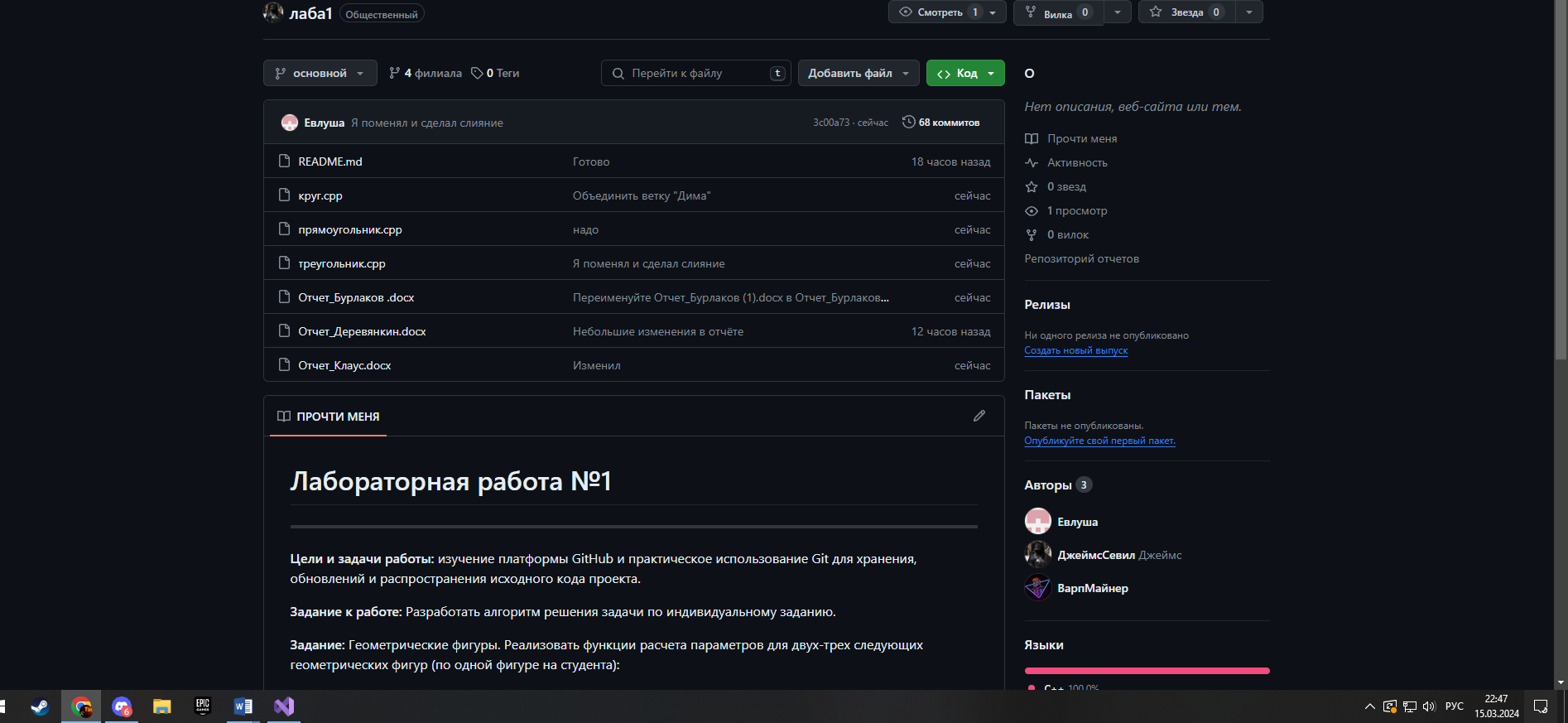
Перед слиянием сохраню ветку и сделаю комментарий



Заходим на ветку main и прописываем команду git merge ilya



Далее проходим на гит и смотрим что изменилось



**Вывод**

В результате работы были выполнены все поставленные цели и задачи: изучена платформа GitHub и практическое использование Git для хранения, обновления и распространения исходного кода проекта. Научились работать над проектом в команде.