

Лабораторная работа №3: Основы работы с Git. Интеграция GitHub с Visual Studio.

Цель работы:

- Знакомство с основными принципами ООП.

Теоретическая информация.

Git - это распределенная система управления версиями (SCM), которая позволяет разработчикам хранить, отслеживать и управлять изменениями в коде и других данных.

Главным преимуществом git-систем является то каким образом хранится информация об изменениях в проекте. Вместо того чтобы хранить каждую из промежуточных версии программы в git-системах хранятся непосредственно изменения между версиями – к примеру какие строки кода отличаются от версии к версии. Таким образом сам проект и все возможные промежуточные версии занимают размер не на много больше чем размер самого проекта.

В git-системах проекты хранятся в репозиториях. **Репозиторием** называется коллекция файлов и директорий, которая используется для управления версиями программного обеспечения или других данных.

Сохранение происходит с помощью коммита. **Коммит** это сохраненное состояние репозитория, которое представляет собой набор изменений, которые были сделаны с момента предыдущего коммита.

Задание №1. Начало работы с GitHub.

Задачи:

- Выполнить действия по созданию репозитория описанные в разделе

В данной работе в качестве git-системы будет использоваться система GitHub. Для начала работы требуется создать аккаунт на следующем сайте: <https://github.com>:

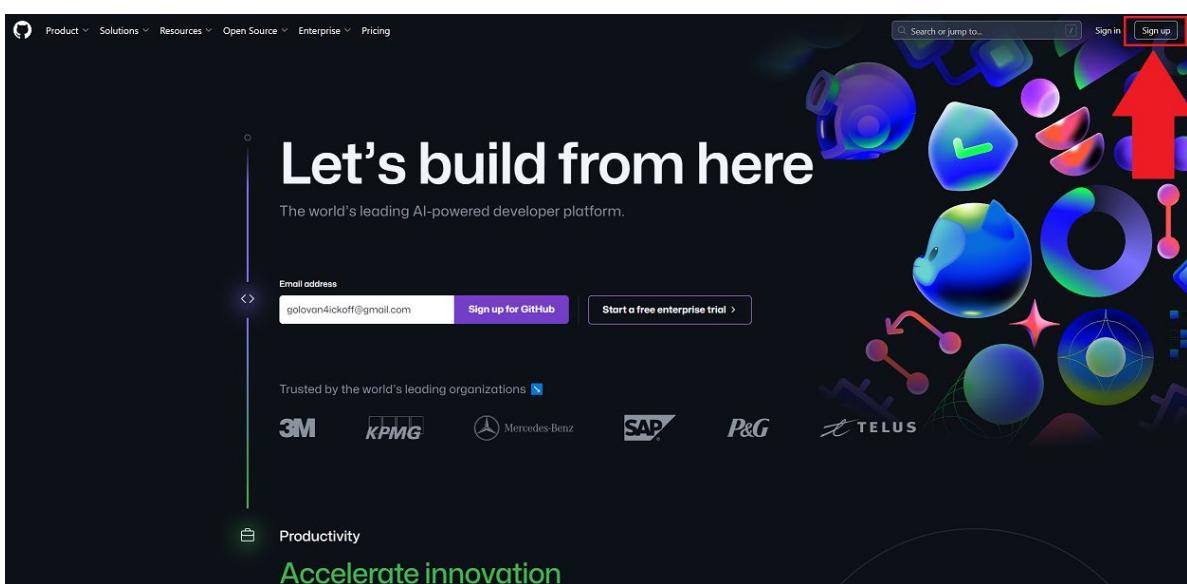


Рисунок 1 – Кнопка регистрации на сайте.

Далее создадим новый проект в Visual Studio. На основе этого проекта будет создан новый репозиторий. Для этого требуется выбрать панель «Git» в верхней панели инструментов Visual Studio. Выберите пункт «Создать репозиторий Git»:

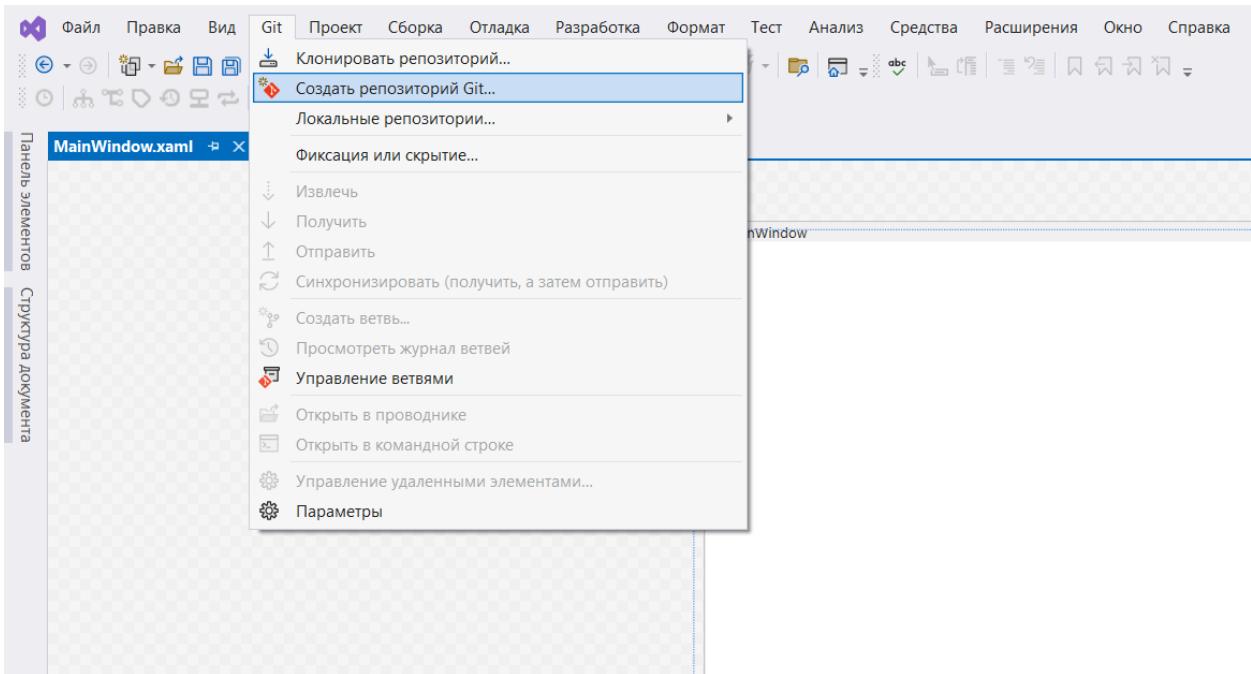


Рисунок 2 – Создание нового репозитория.

В открывшемся окне потребуется выбрать аккаунт GitHub в котором будет создан новый репозиторий. После выбора пункта добавления новой учетной записи войдите в свой аккаунт GitHub. После этого нажмите «Создать и отправить». После этого автоматически будет произведен первый коммит и весь проект будет сохранен в удаленном репозитории GitHub, но так же будет создан локальный репозиторий на компьютере где был создан проект.

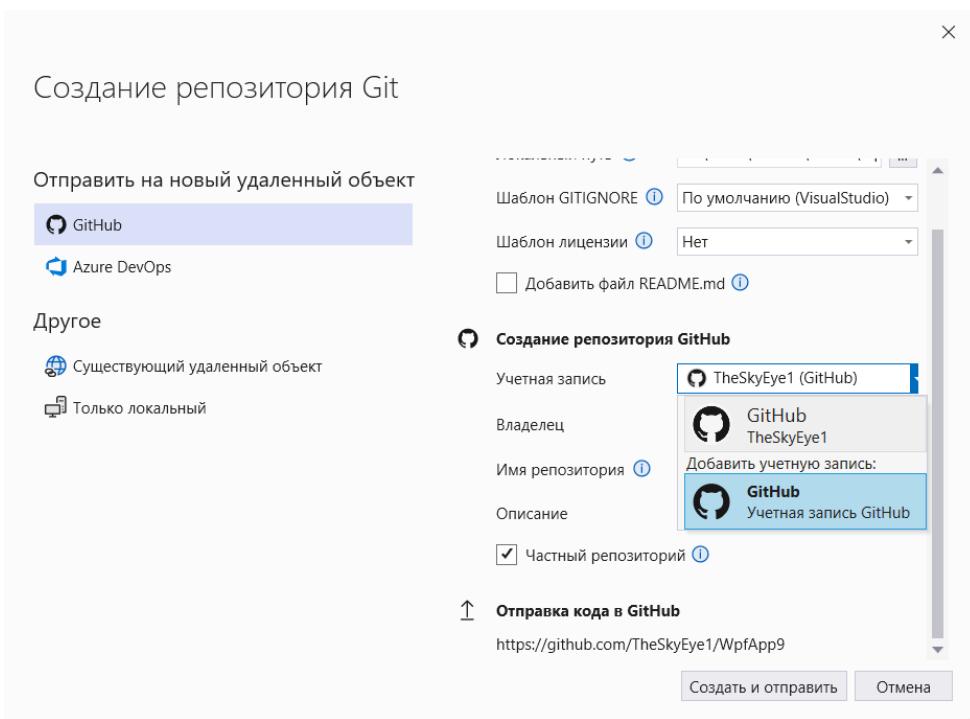


Рисунок 3 – Добавление новой учетной записи.

В Visual Studio встроены удобные инструменты для работы с git-системой. В окне «Вид» можно выбрать пункт «Изменения Git». Данная панель используется для выполнения и создания коммитов.

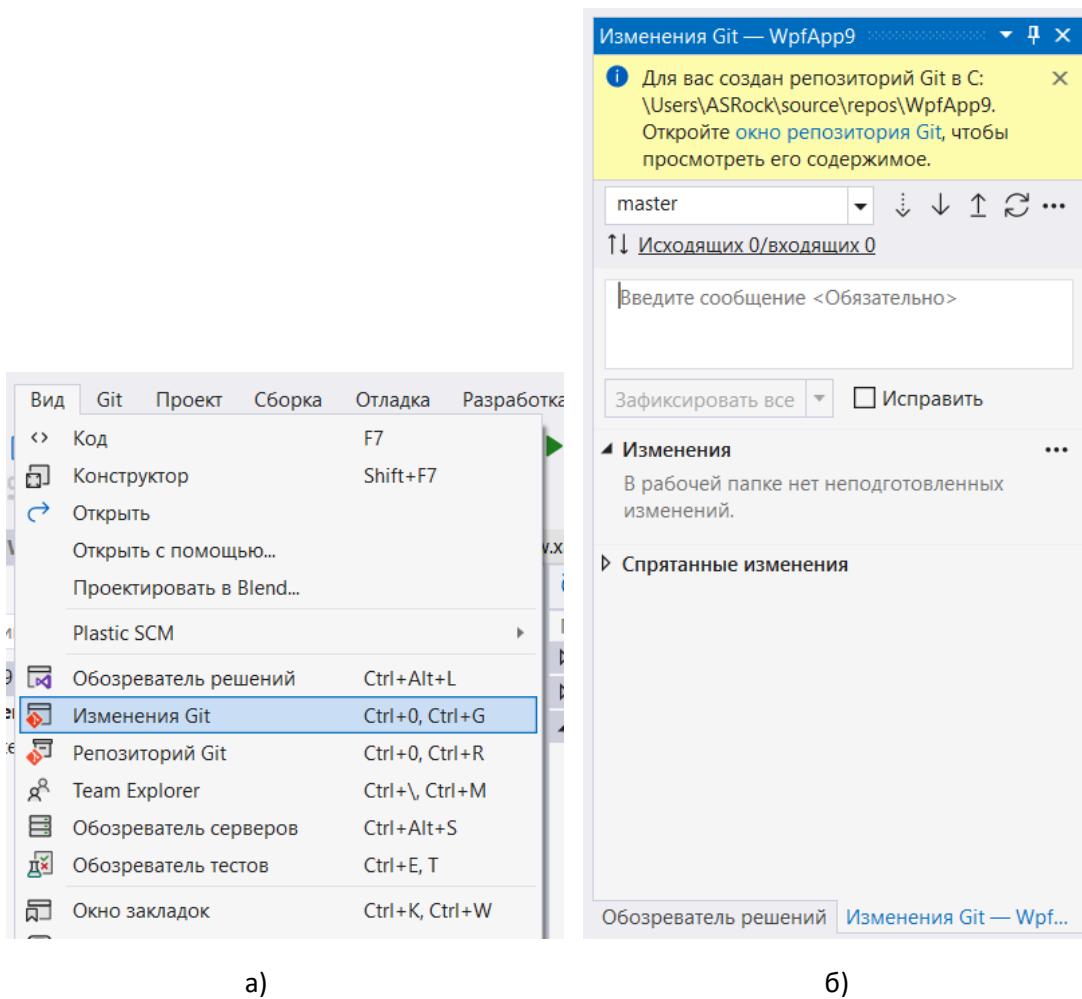


Рисунок 4 – Панель изменений Git: а) открытие панели изменений Git;
б) панель изменений git.

Для того чтобы создать коммит можно добавить любую строку кода в код программы. Затем в окне изменений Git требуется написать комментарий к коммиту. Если все было выполнено корректно – кнопка «Зафиксировать все» должна быть активной. По нажатию на данную кнопку будет создан коммит. Но данный коммит будет только локальным, то есть будет сохранен только в локальном репозитории. Для того чтобы отправить коммит в удаленный репозиторий требуется нажать кнопку «Отправить» (стрелка направленная вверх).

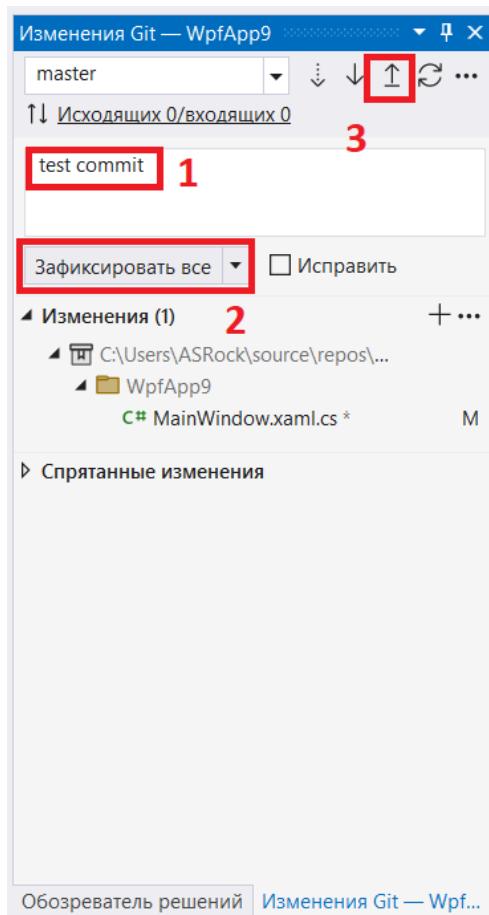


Рисунок 5 – Создание коммита через панель изменений: 1) поле ввода комментария; 2) кнопка создания коммита; 3) отправление коммита в удаленный репозиторий.

После выполнения этих операций проверить сохранение коммита на локальном репозитории можно в окне «Репозиторий Git» (кнопка открытия находится под кнопкой открытия панели изменений git).

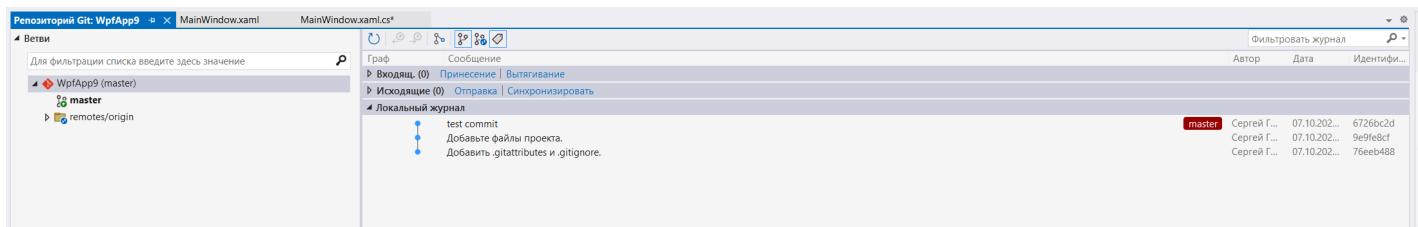


Рисунок 6 – Список изменений в репозитории.

Для просмотра удаленного репозитория можно зайти на GitHub, все доступные репозитории должны отображаться в профиле.

Задание №2. Операции с ветвями в Git.

Разработать простой калькулятор на WPF. Каждая из арифметических операций должна быть реализована в отдельной ветви git-проекта, финальный проект должен быть слиянием всех реализованных ветвей.

Функционал программы:

- Калькулятор содержащий два поля в который пользователь может ввести свои значения;
- Кнопки, к которым привязаны арифметические операции: сложение, вычитание, умножение, деление;
- По нажатию кнопки в поле вывода выводится результат арифметической операции между введёнными значениями.

Задачи:

- Реализовать арифметические функции и связанный с ними интерфейс в отдельных ветвях.
- Провести слияние ветвей в финальный проект.
- Ветви должны быть синхронизированы с удаленным репозиторием.

Для создания новой ветви воспользуйтесь окном изменений. Выберите выпадающий список с ветвями, в открывшемся окне выберите «Создать новую ветвь»:

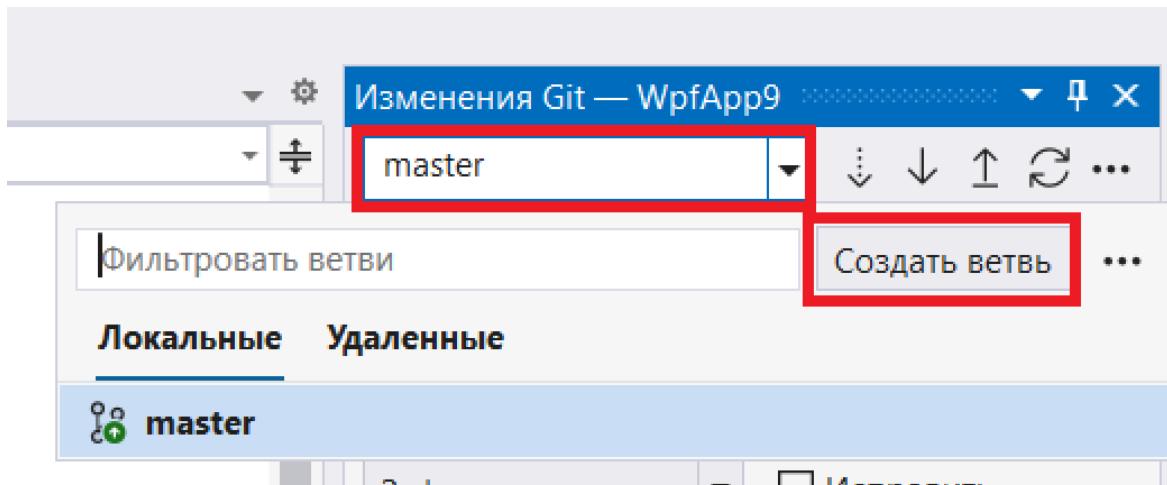


Рисунок 7 – Список ветвей проекта.

В открывшемся окне требуется ввести название новой ветви, а также выбрать от какой ветви будет происходить наследование:

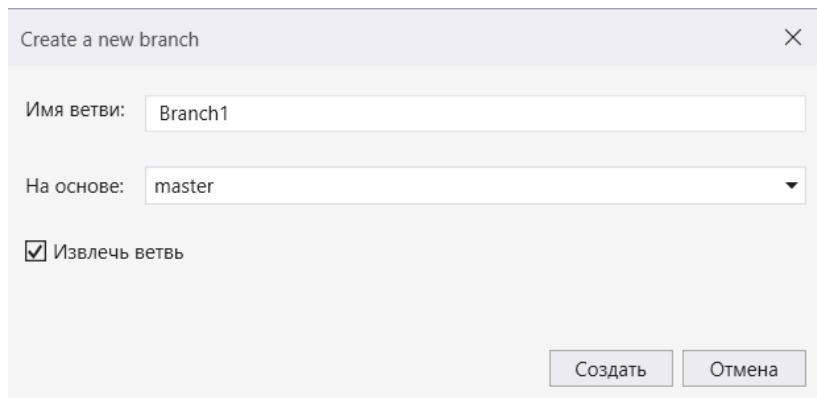


Рисунок 8 – Окно создание ветвей.

Переключаться между ветвями можно через выпадающий список в панели изменений.

Зачастую ветви используются для создания и тестирования нового функционала, для того чтобы не нарушить работоспособность основной программы. Но если новый код был протестирован и его нужно перенести в основную программу используется слияние ветвей.

Для того чтобы объединить две ветви сначала откройте ветвь, в которую будет происходить слияние. После этого перейдите в меню «Репозиторий Git» и выберите в списке ветвей ту с которой должно происходить слияние. В выпавшем меню выберите пункт «Объединить ветвь»:

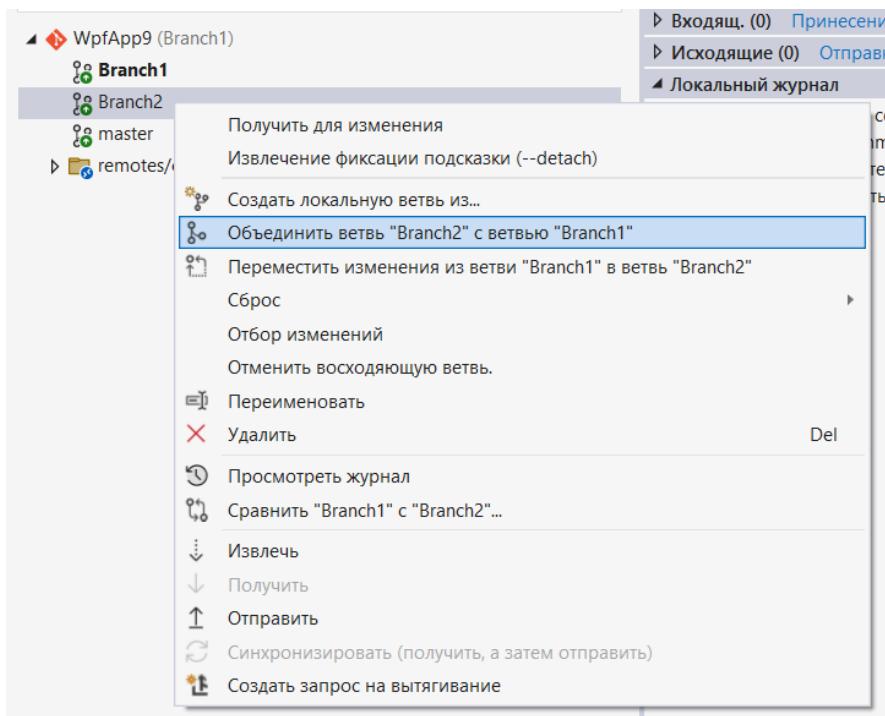


Рисунок 9 – Окно выбора ветвей для объединения.

Вероятнее всего ничего не произойдет – это связано с тем что при объединении произошли конфликты, которые требуется решить.

Для этого перейдите в окно, в котором компилятор показывает ошибки слияния. В данном окне потребуется открыть редактор слияния, для этого нажмите на ссылку в выделенном предупреждении:

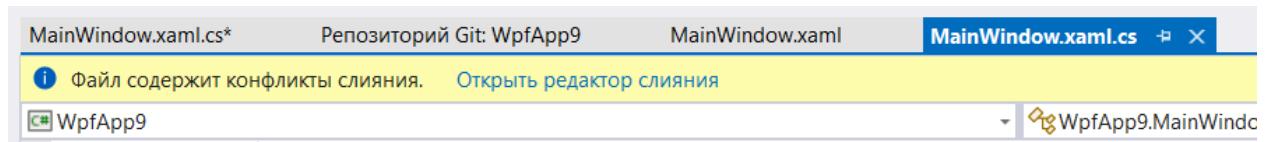


Рисунок 10 – Предупреждение с редактором слияния.

В появившемся окне вы можете выбрать какая часть, когда из какой ветви будет присутствовать в финальном слиянии: можно выбрать как код только из одной ветви, так и из обеих. После того как конфликты были решены нажмите «Принять слияние»:

This screenshot shows the 'Git Merge Editor' in Visual Studio. It displays two versions of the same code file, 'MainWindow.xaml.cs'. The left pane shows code from 'Branch2' with a conflict at line 29: 'double k = 0.24; double q = 1.45;'. The right pane shows code from 'Branch1' with a conflict at line 30: 'int c = 2; int d = 3; int f = a + b + c + d;'. The bottom pane shows the resulting merged code where both lines are present. A status bar at the bottom indicates 'Стр: 28 Симв: 1 Проблемы CRLF'.

```

MainWindow.xaml.cs*  Репозиторий Git: WpfApp9  MainWindow.xaml  MainWindow.xaml.cs
↑ ↓ Показать только конфликты Принять вхождение Принять текущие Сравнить Принять слияние Показать различия слов
[...] Конфликты: 1 (Осталось: 0)
 Входящие: Branch2
22
23     public MainWindow()
24     {
25         InitializeComponent();
26         int a = 10;
27         int b = 5;
28         double k = 0.24;
29         double q = 1.45;
30     }
31 }
32 }
33
34
Результат: WpfApp9/MainWindow.xaml.cs
22
23     public MainWindow()
24     {
25         InitializeComponent();
26         int a = 10;
27         int b = 5;
28         int c = 2;
29         int d = 3;
30         int f = a + b + c + d;
31         double k = 0.24;
32         double q = 1.45;
33     }
34 }
35
36
Стр: 28 Симв: 1 Проблемы CRLF
Стр: 33 Симв: 1 Проблемы CRLF

```

Рисунок 11 – Окно редактирования конфликтов.

После совершения слияния стоит совершить коммит результата слияния. В окне «Репозитория Git» можно увидеть развитие структуры проекта, то как происходит ветвление и слияние веток:

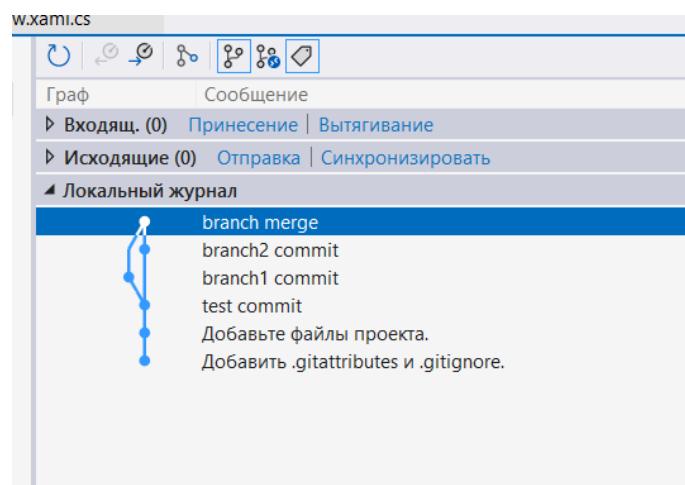


Рисунок 12 – Результат слияния в структуре проекта.

Справочная информация.

Одной из важнейших функций git-систем является возможность вернуться к предыдущей сохраненной итерации. Для этого выберите один из коммитов в структуре проекта, а затем выберите функцию «Извлечь»:

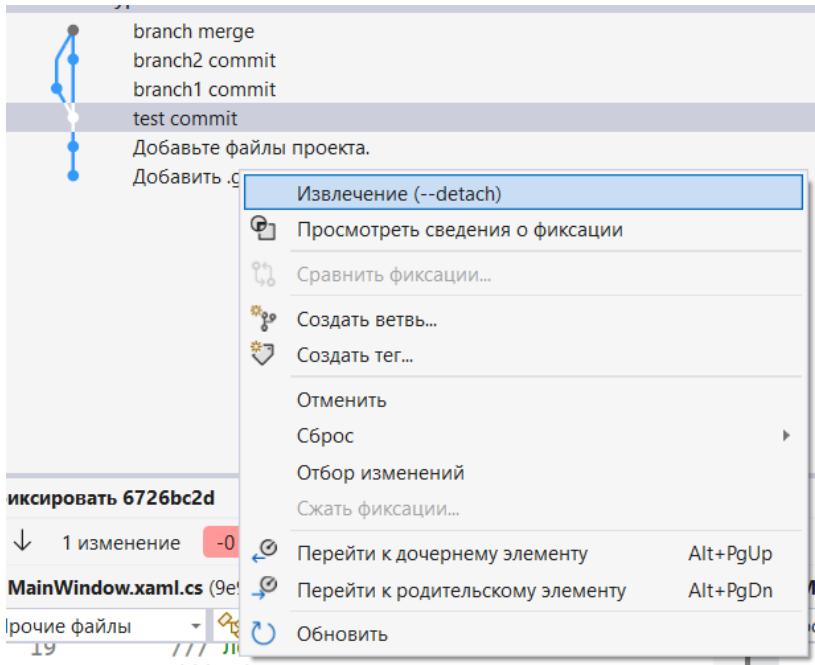


Рисунок 13 – Извлечение предыдущей сохраненной версии проекта.

Удаленный репозиторий можно свободно перемещать на другие компьютеры. Для того чтобы скопировать проект на другом компьютере при открытии Visual Studio выберите пункт «Клонирование репозитория», после чего потребуется передать ссылку на проект. Ссылку на проект можно найти в репозитории на GitHub:

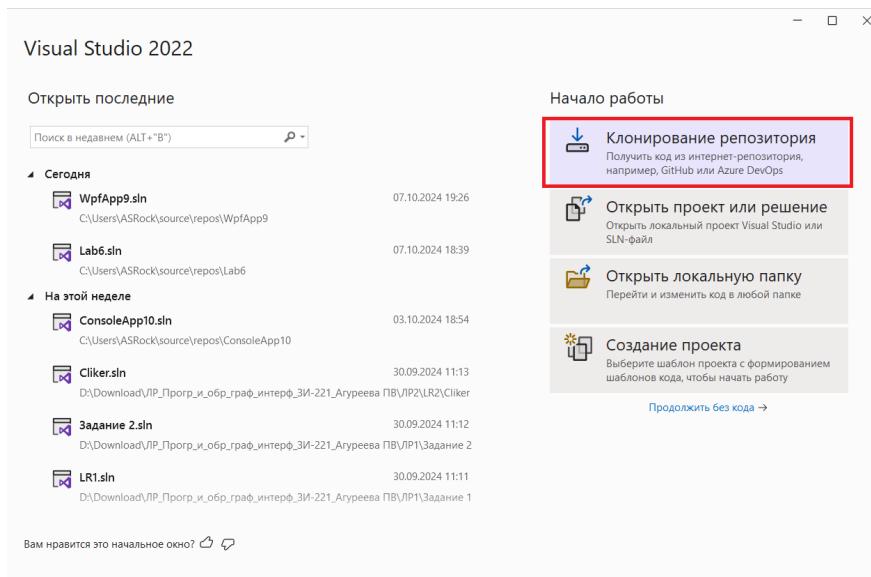


Рисунок 14 – Клонирование репозитория

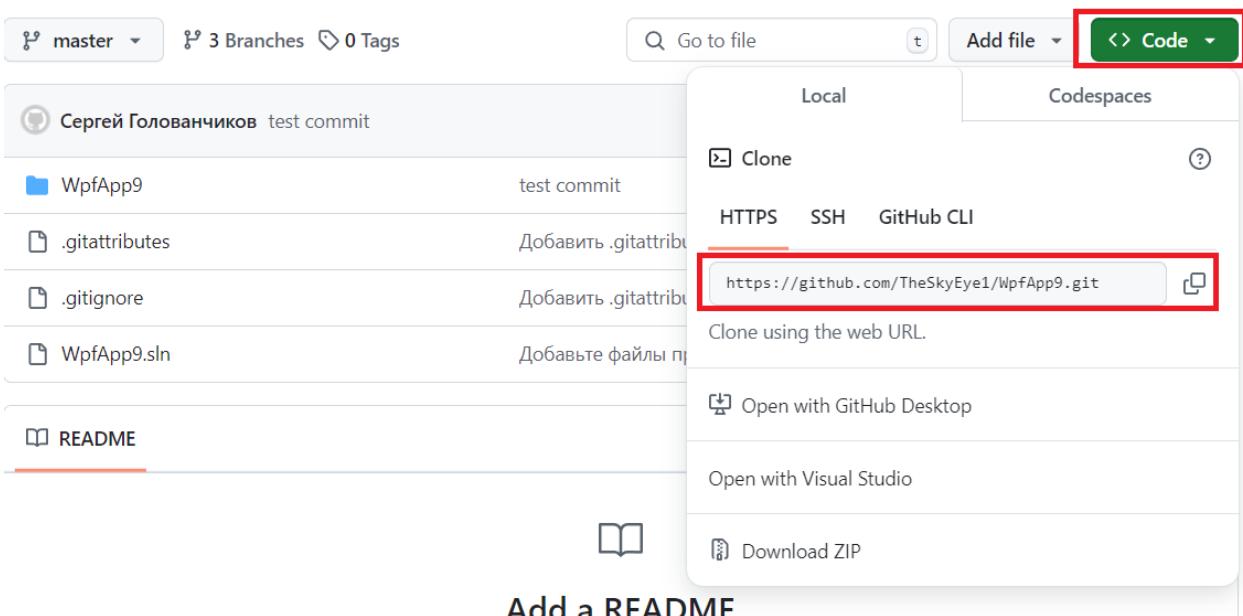


Рисунок 15 – Ссылка на проект в репозитории на GitHub