***Clone***

*Clone* – функция, которая клонирует репозиторий в новый каталог, создает удалённо отслеживаемые ветки для каждой ветки в клонированном репозитории, создаёт и извлекает начальную ветку, которая ответвляется от текущей активной ветки клонированного репозитория [https://git-scm.com/docs/git-clone]. Схема функции *git clone* представлена на Рисунке 1.

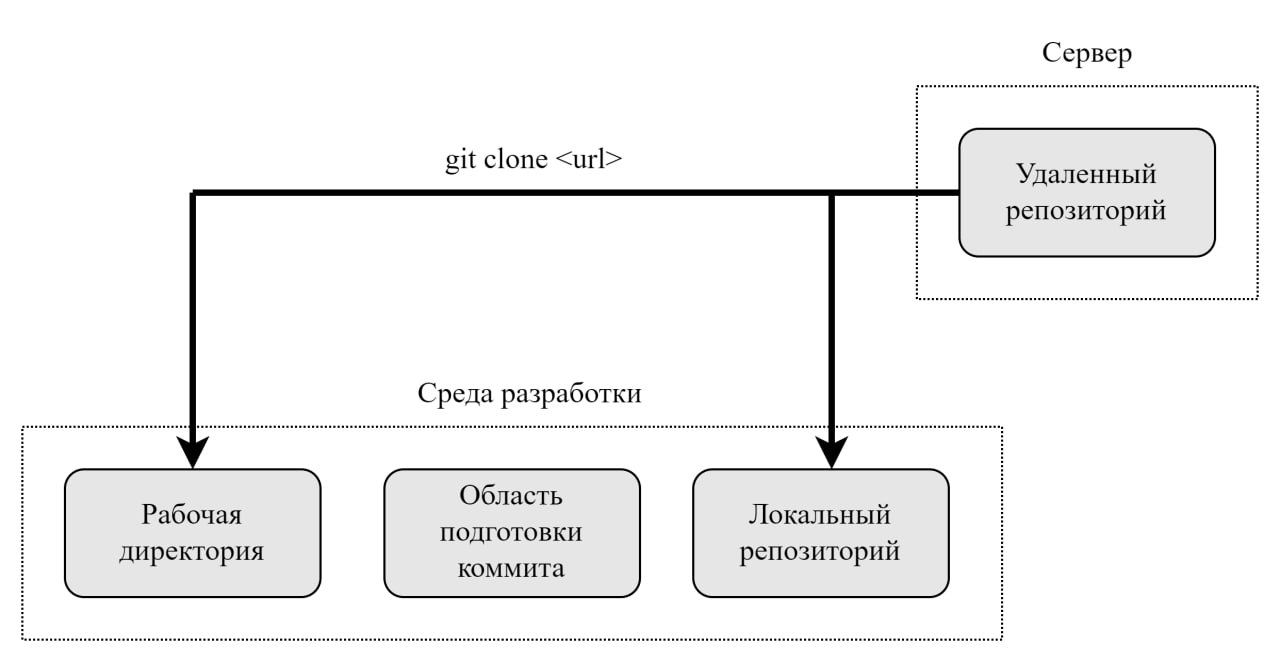


Рисунок 1 – Схема функции *git clone*

**Функция *сlone* в *Visual Studio* 2022**

Для того, чтобы воспользоваться функцией *clone*, необходимо перейти во вкладку *Git* в *Visual Studio 2022*. Вкладка *Git* на Рисунке 2.

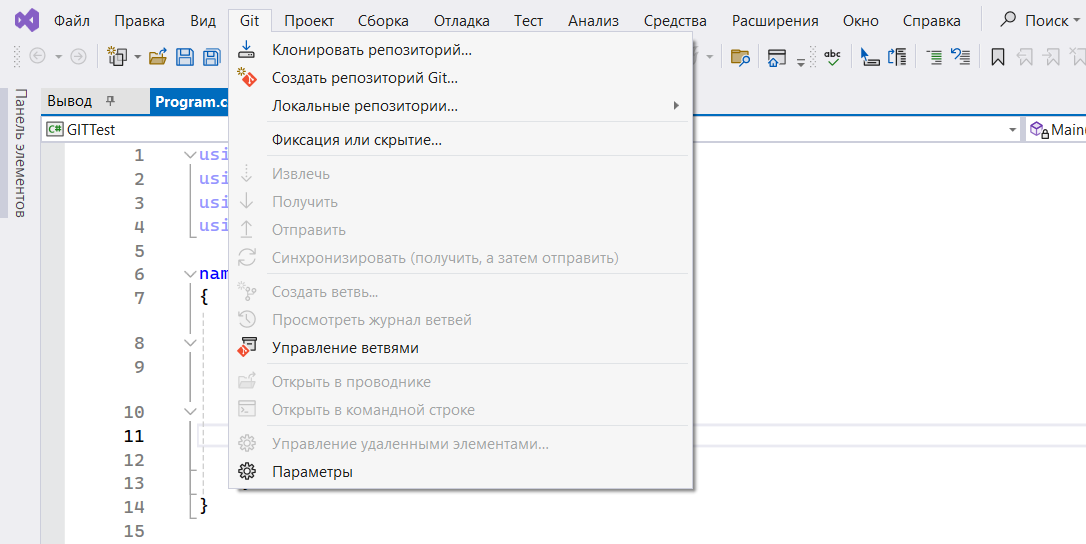
**

Рисунок 2 – Вкладка *Git*

После нажатия на кнопку *Git*, появится выпадающий список, в котором необходимо перейти в раздел «Клонировать репозиторий…». Раздел «Клонировать репозиторий…» на Рисунке 3.

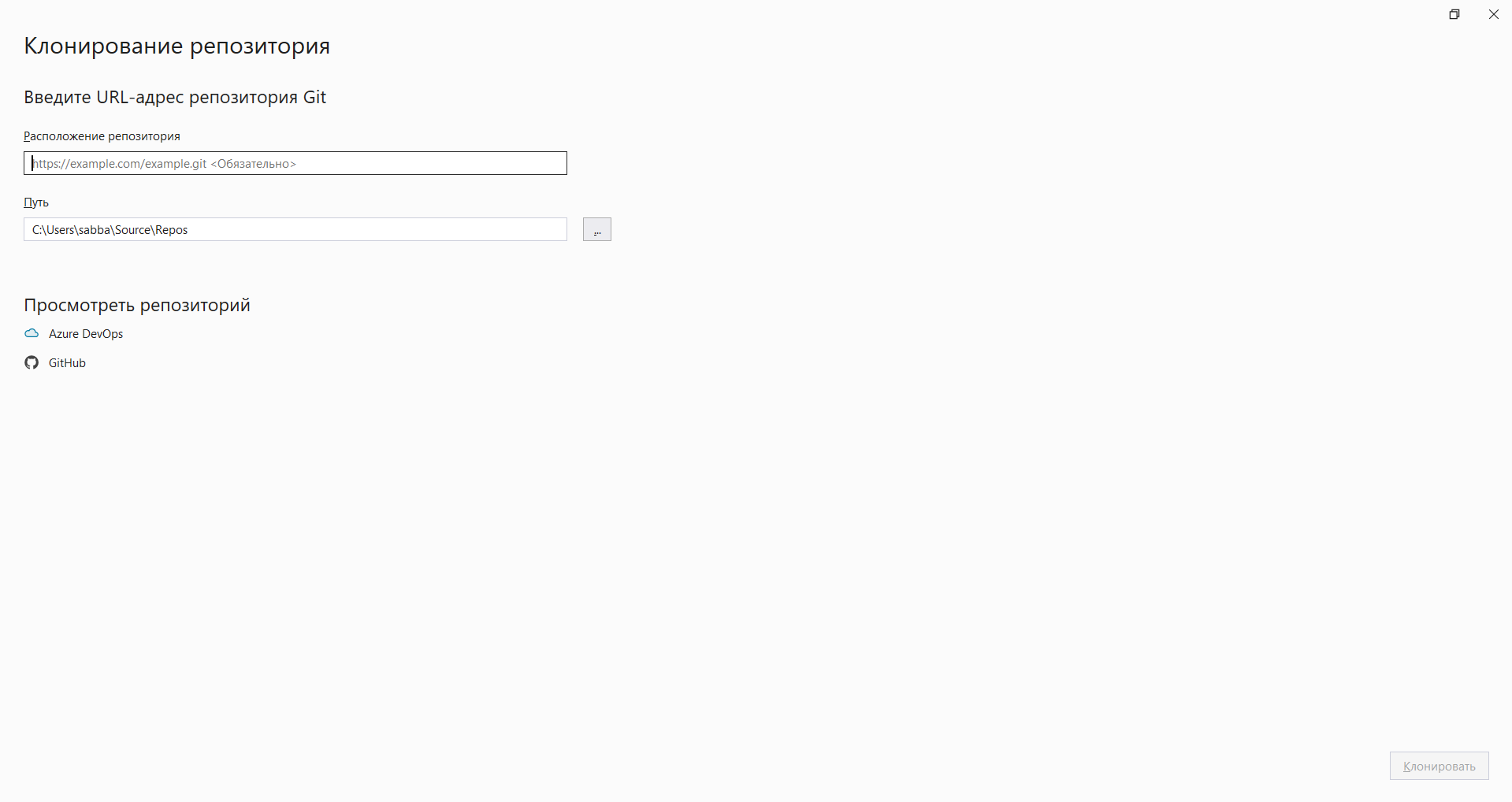
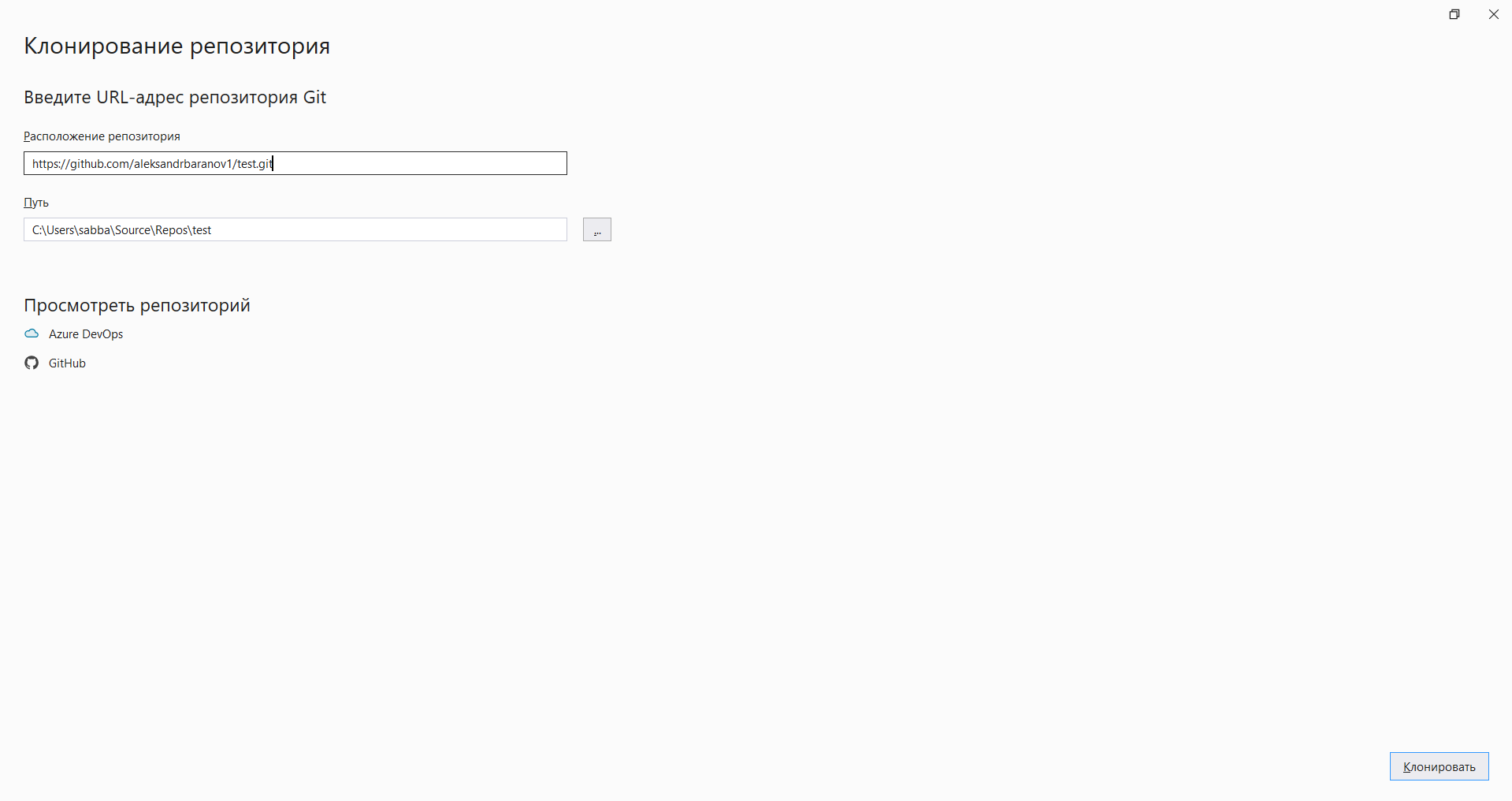


Рисунок 3 – Раздел «Клонировать репозиторий…»

В текстовое поле «Расположение репозитория» необходимо вставить ссылку на удаленный репозиторий. В текстовом поле «Путь» необходимо выбрать расположение, где будет находиться клонированный репозиторий. Пример заполнения полей раздела «Клонирование репозитория» на Рисунке 4.



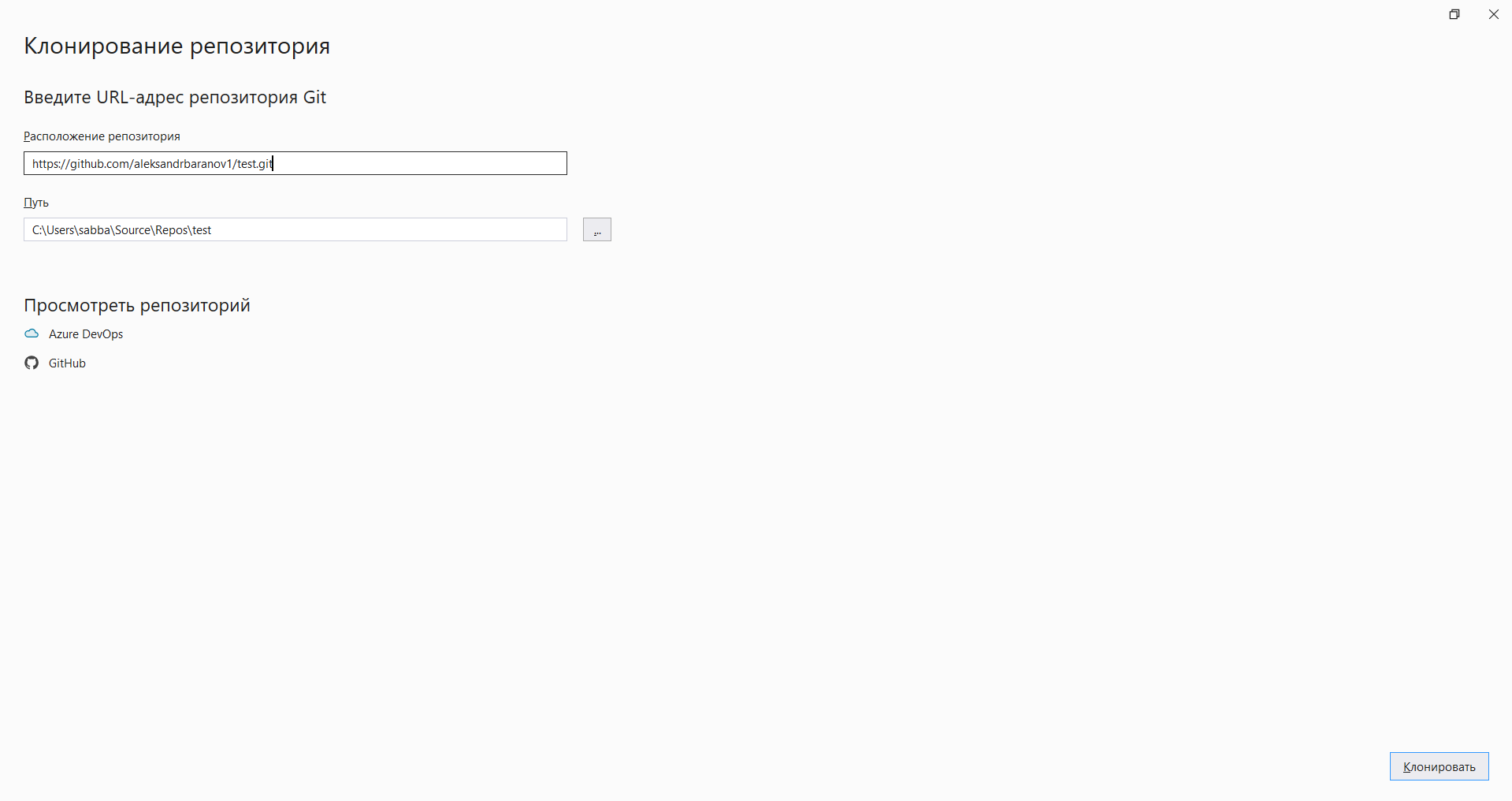


Рисунок 4 – Раздела «Клонирование репозитория»

После Заполнения необходимо нажать кнопку «Клонировать», после чего удаленный репозиторий появится локально на компьютере. Клонированный репозиторий на Рисунке 5.

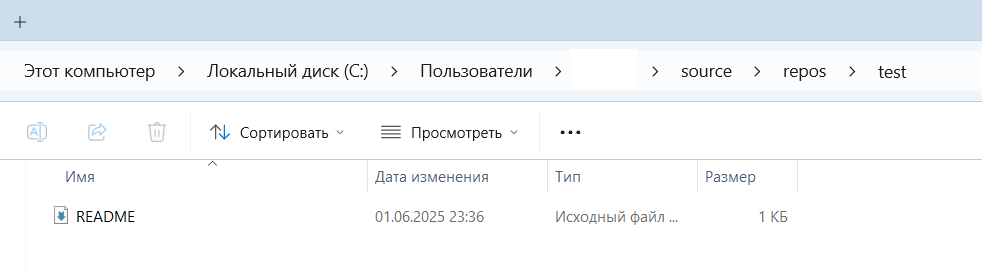


Рисунок 5 – Клонированный репозиторий

***Commit***

*Commit* (коммит)– функция, которая создаёт новый коммит, содержащий текущее содержимое индекса (текущие изменения в репозитории) и заданное сообщение, которое описывает изменения. Новый коммит является прямым потомком *HEAD*, обычно это последний коммит текущей ветки, и ветка обновляется, чтобы указывать на него [https://git-scm.com/docs/git-commit]. Принцип работы функции *git commit* на Рисунке 6.

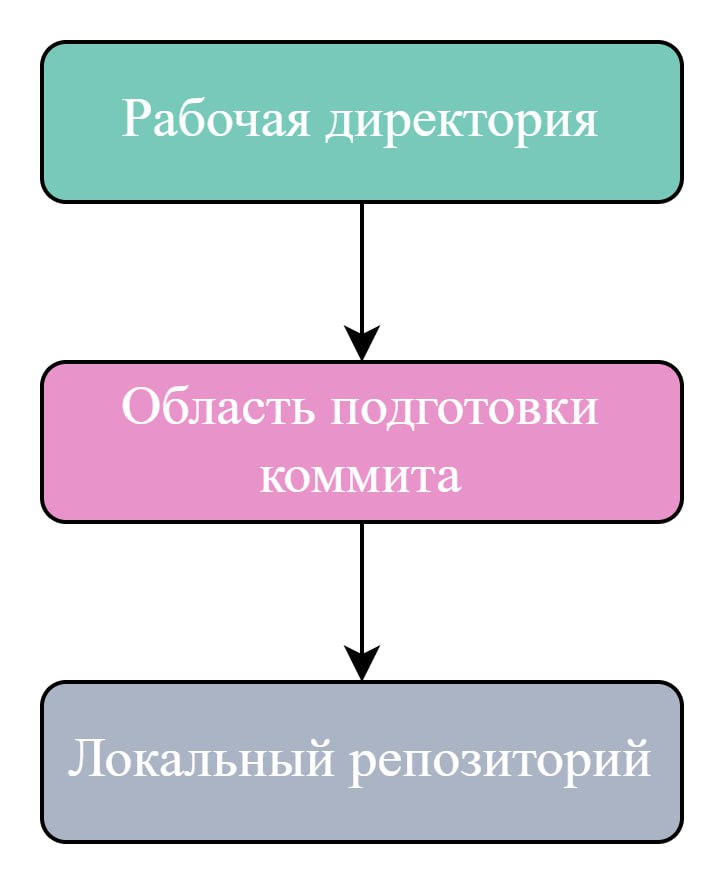


Рисунок 6 – Принцип работы функции *git commit*

**Функция *commit* в *Visual Studio* 2022**

Для того, чтобы воспользоваться функцией *commit*, не обходимо из выпадающего списка *Git* перейти в раздел «Фиксация или скрытие…». Раздел «Фиксация или скрытие…» на Рисунке 7.

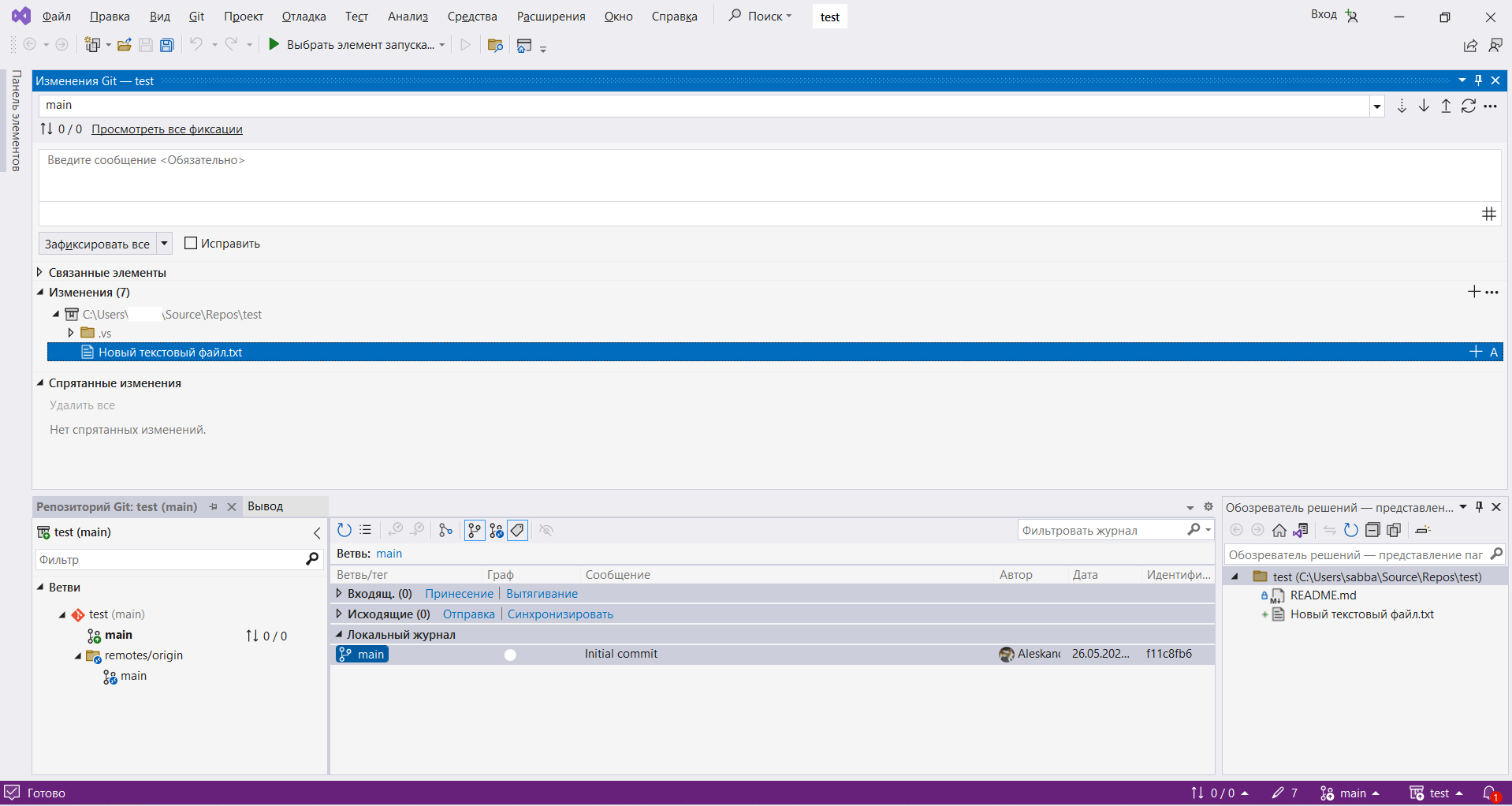


Рисунок 7 – Раздел «Фиксация или скрытие…»

Для того чтобы добавить изменения в индекс коммита, необходимо выделить те файлы, которые необходимо добавить в индекс, нажать на по ним правой кнопкой мыши и в появившемся списке выбрать «Промежуточное хранение». Альтернативным способ добавления изменений в индекс является нажатие значка «+» справа от файла. Функция «Промежуточное хранение» на Рисунке 8.

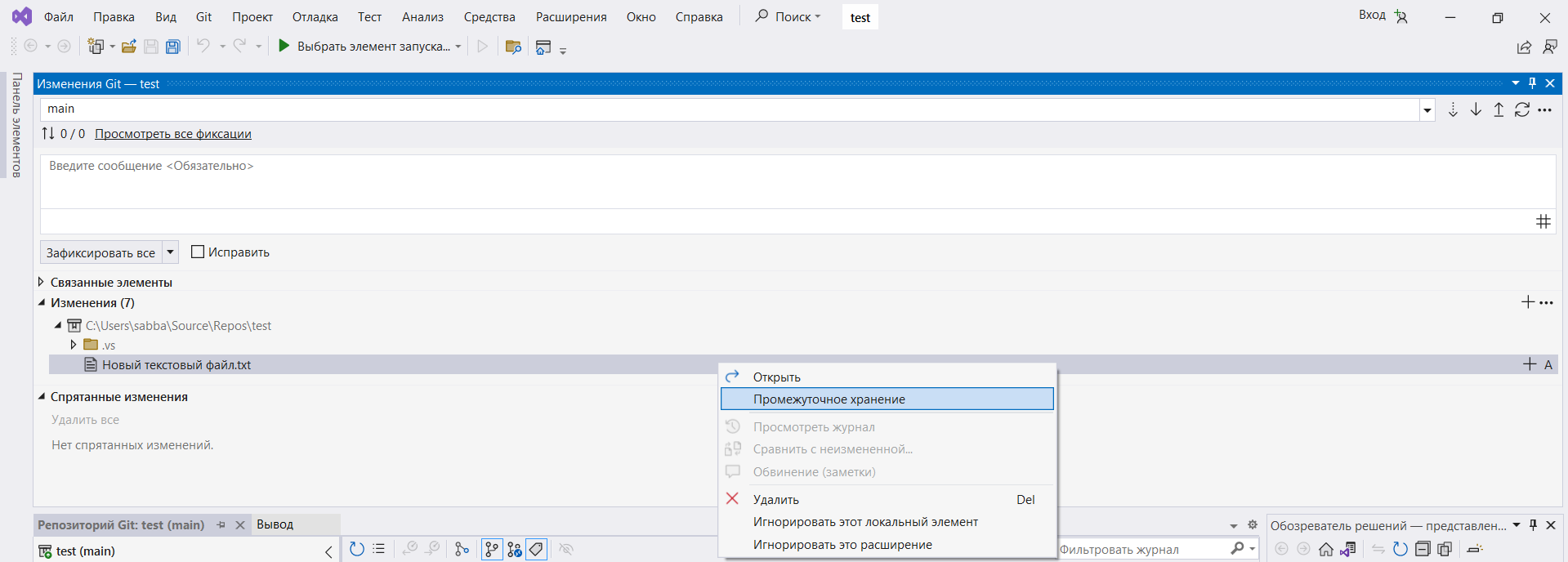


Рисунок 8 – Раздел «Фиксация или скрытие…»

После добавления изменений в индекс, необходимо написать сообщение, описывающее добавленные изменения. Для этого необходимо ввести сообщение в поле «Введите сообщение». Пример сообщения, которое описывает изменения на Рисунке 9.

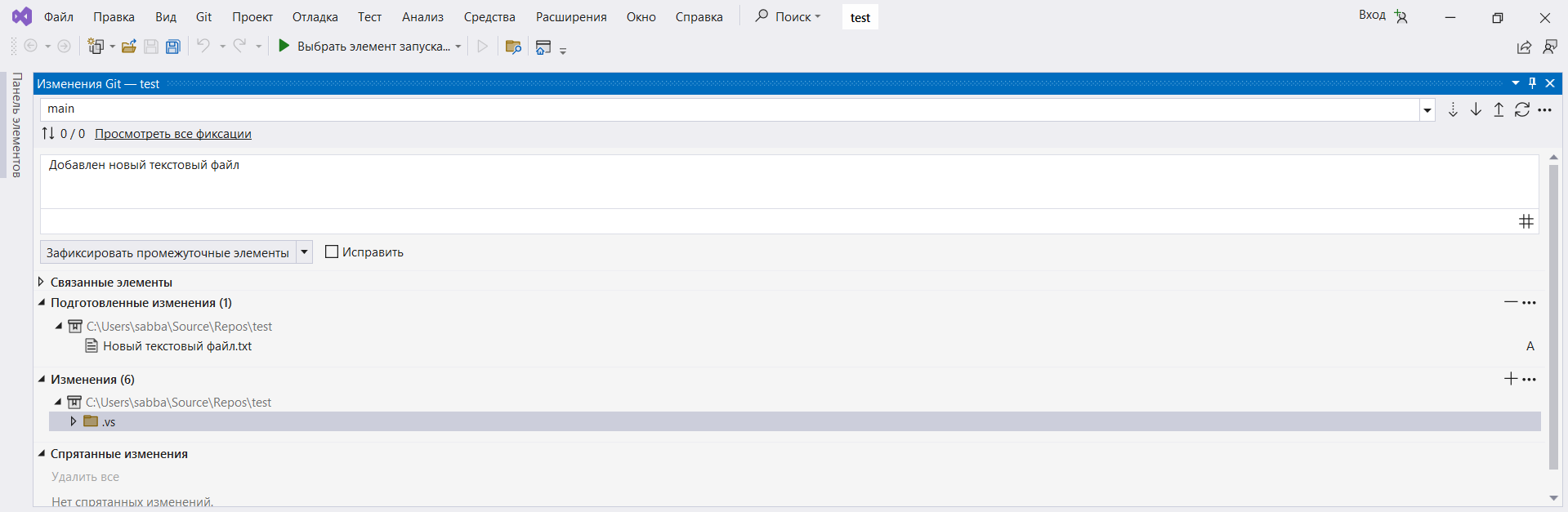


Рисунок 9 – Пример сообщения, которое описывает изменения

Чтобы создать коммит необходимо нажать на кнопку «Зафиксировать промежуточные изменения», после чего появится сообщение о успешно созданном коммите. Успешно созданный коммит на Рисунке 10.

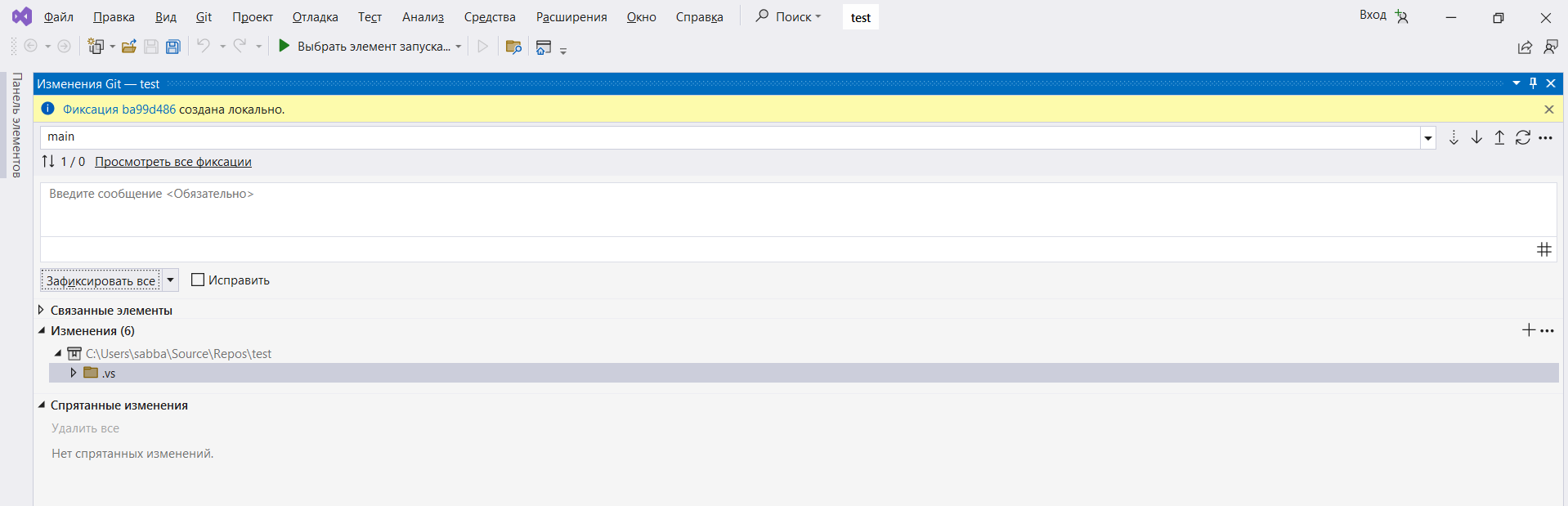
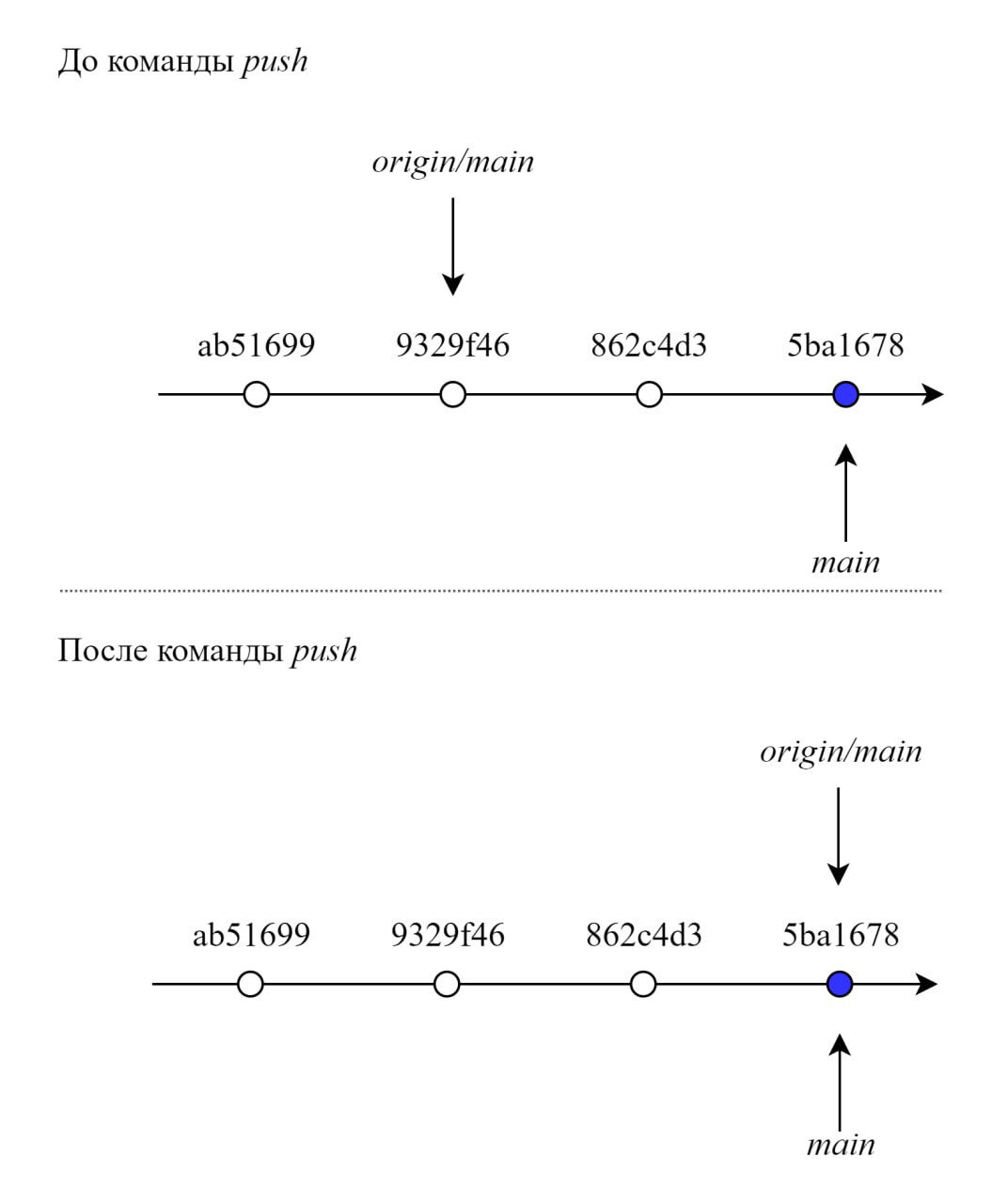


Рисунок 10 – Успешно созданный коммит

***Push***

*Push* – функция обновления удалённых веток, которая использует локальные ветки, отправляя объекты, необходимые для завершения указанных веток [https://git-scm.com/docs/git-push]. Принцип работы функции *git* *push* на Рисунке 11.



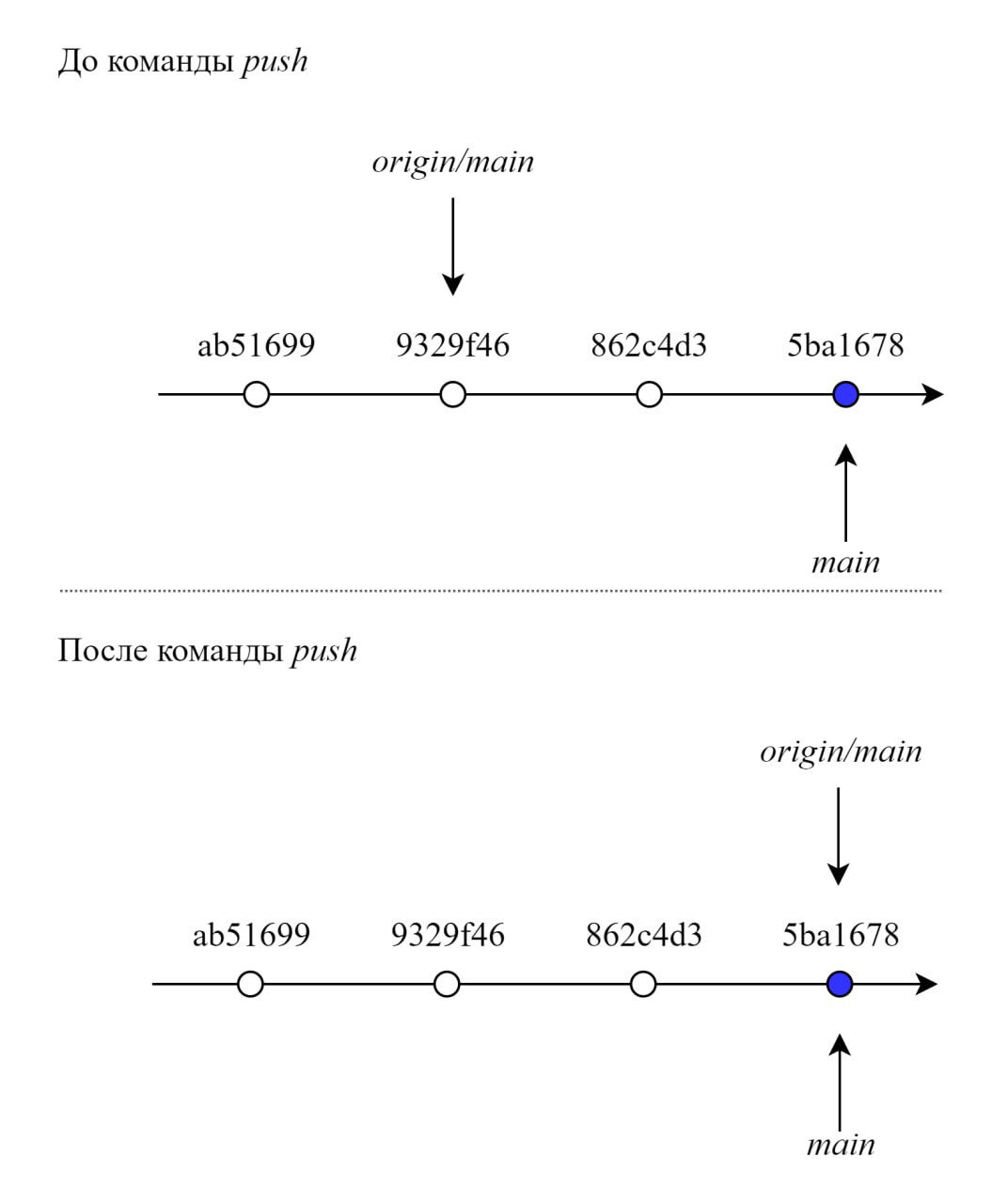


Рисунок 11 – Принцип работы функции *git push*

**Функция *push* в *Visual Studio* 2022**

Для того, чтобы воспользоваться функцией *Push*, необходимо из выпадающего списка *Git* перейти в раздел «Отправить». После нажатия кнопки «Отправить», локальные изменения будут отправлены ы удаленный репозиторий. Коммиты в удаленном репозитории после отправки на Рисунке 12.

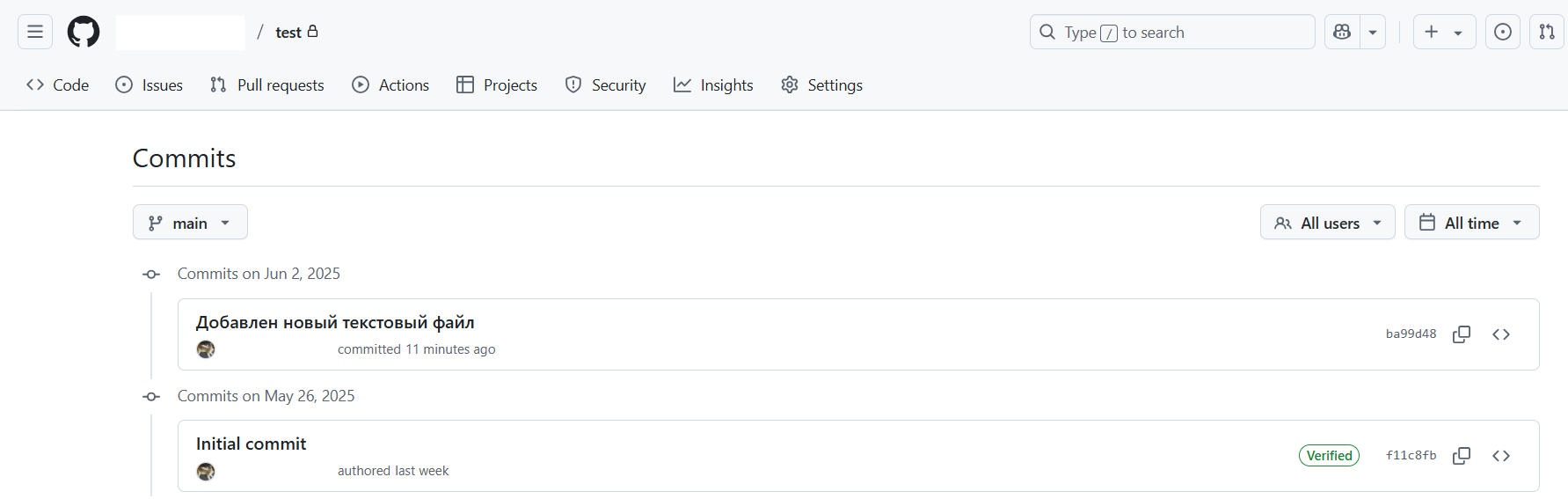


Рисунок 12 – Коммиты в удаленном репозитории после отправки

***Merge***

*Merge* (слияние) – это процесс объединения изменений из одной ветки в другую. Принцип работы функции *git merge* на Рисунке 13.

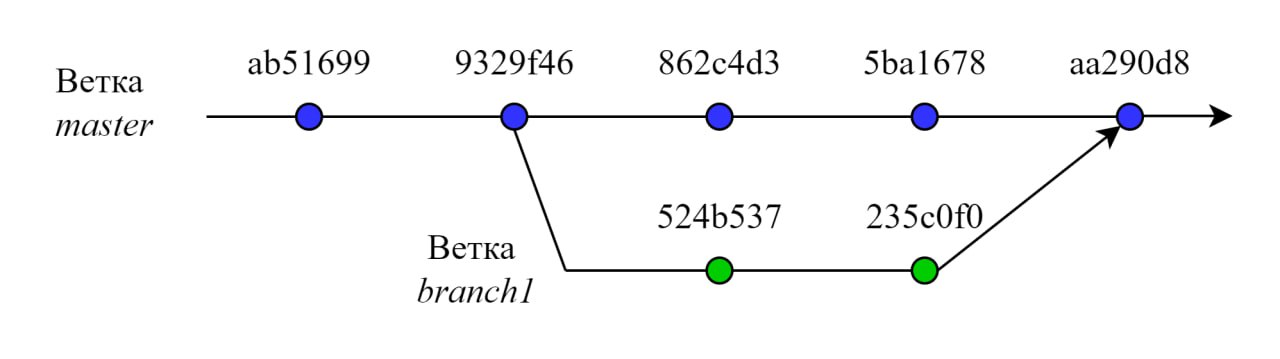
**

Рисунок 13 – Принцип работы функции *git merge*

**Функция *merge* в *Visual Studio* 2022**

Для того, чтобы воспользоваться функцией *merge*, необходимо находиться в той ветке, в которую необходимо сделать слияние. Чтобы перемещаться между ветками, необходимо нажать комбинацию клавиш *Ctrl*+*Alt+F3*, после чего появится окно управления ветками, где нужно выбрать ту ветвь, в которую необходимо сделать слияние. Окно управления ветками на Рисунке 14.

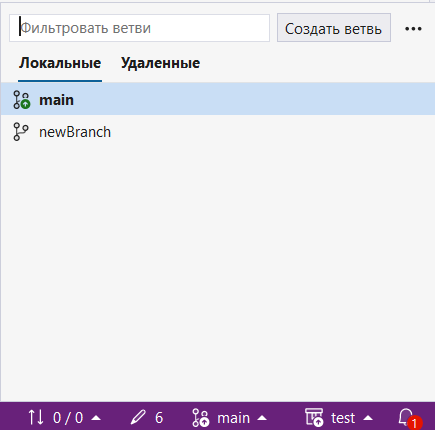


Рисунок 14 – Окно управления ветками

После выбора ветки, необходимо нажать правой кнопкой мыши по ветке, которая будет сливаться и в появившемся списке нажать на кнопку «Объединить ветвь <Название ветки> с ветвью <Название ветки>». Кнопка слияния веток на Рисунке 15.

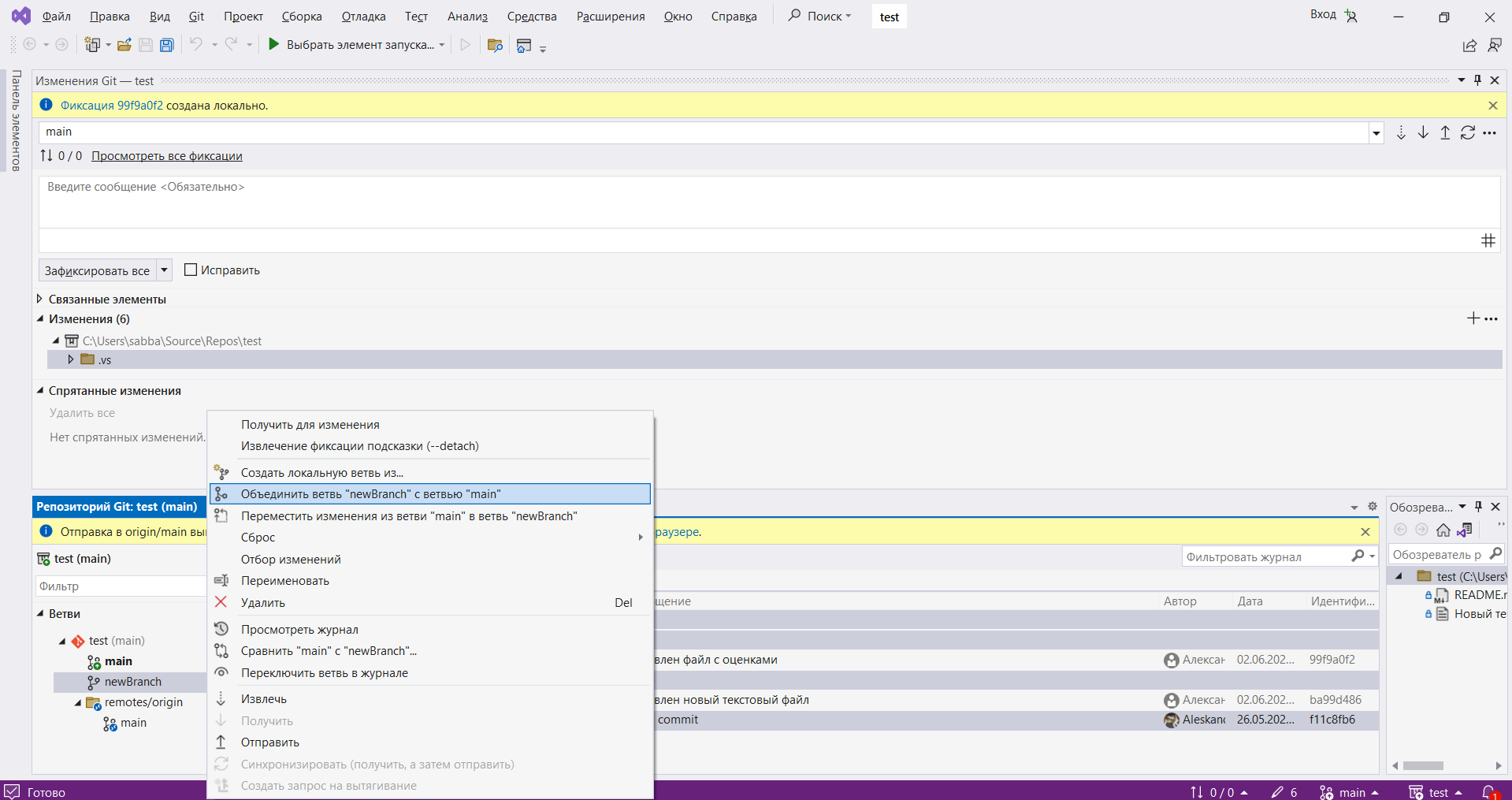
******

Рисунок 15 – Кнопка слияния веток

После нажатия кнопки слияния веток появится окно подтверждения объединения, где необходимо его подтвердить, нажав кнопку «Объединить». Окно подтверждения объединения на Рисунке 16.

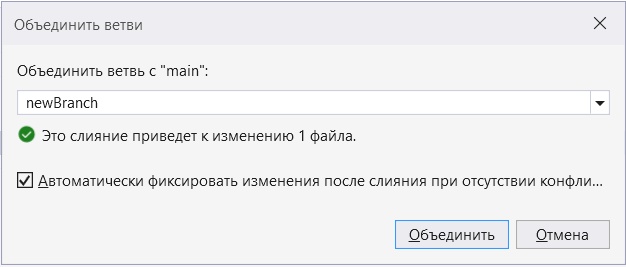


Рисунок 16 – Окно подтверждения объединения

***Fetch***

*Fetch –* функция, которая загружает коммиты, файлы и ветки из удаленного репозитория в локальный. *Fetch* используется, чтобы просмотреть все коммиты во всех ветках. Принцип работы функции *git fetch* на Рисунке 5.

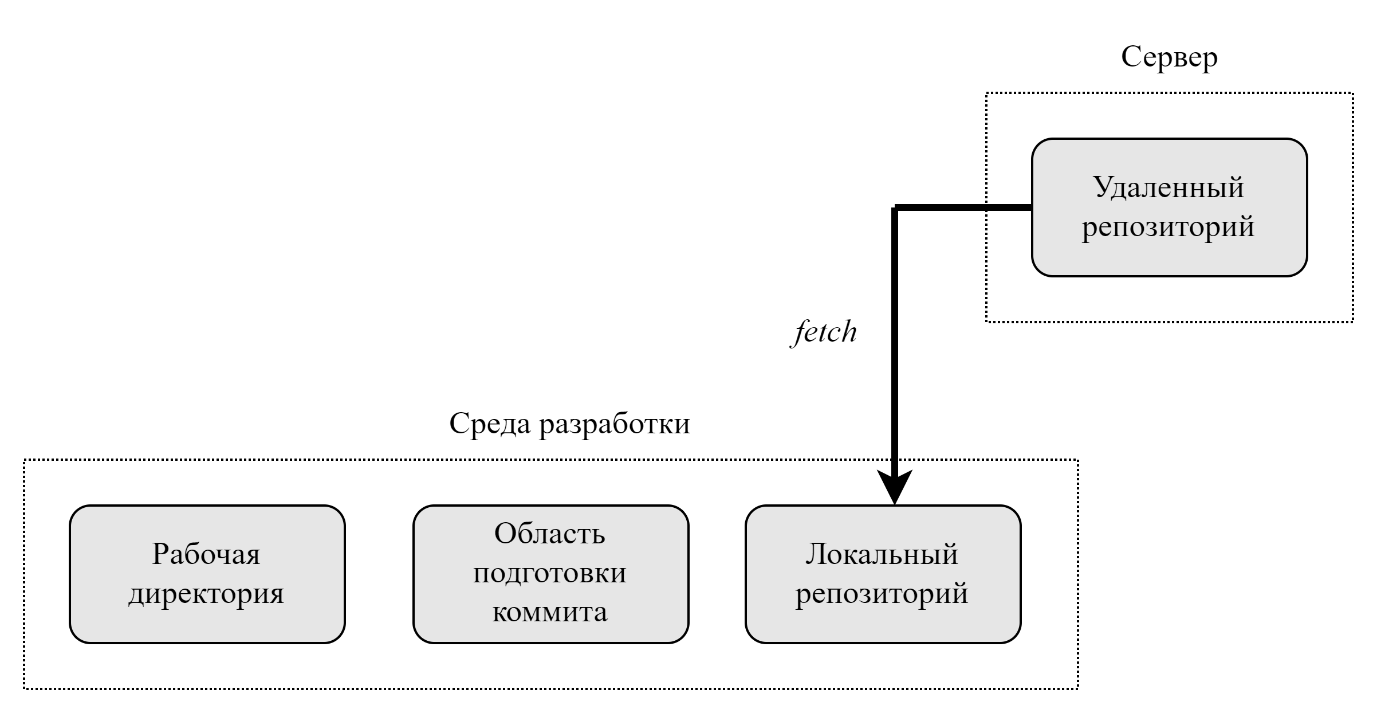
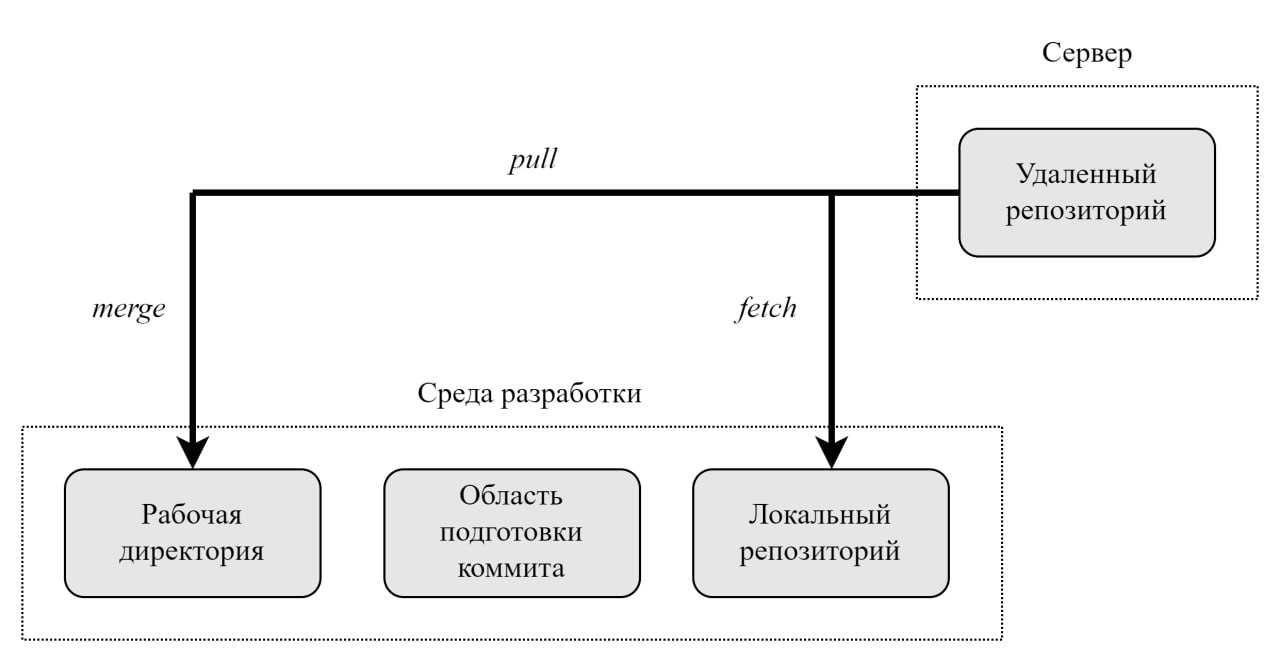


Рисунок 5 – Принцип работы функции *git fetch*

***Pull***

*Pull –* функция, которая включает изменения из удалённого репозитория в текущую ветку. Если текущая ветка находится позади удалённой, то по умолчанию она перематывает текущую ветку вперёд, чтобы она соответствовала удалённой. Пример работы функции *pull* на Рисунке 6.

Рисунок 6 – Пример работы функции *pull*