



Универсальный редактор с открытым исходным кодом

Чагин Ф.С

Карпов Р.В.

Ефимов С.Р.

# Visual Studio Code

Универсальный инструмент для повседневной работы программиста

**Visual Studio Code (VS Code)** – бесплатный кроссплатформенный редактор кода, разработанный Microsoft. Он серьёзно упрощает создание, отладку и поддержку программного обеспечения. Благодаря своей доступности он является популярным выбором для студентов и начинающих разработчиков, а мощная экосистема расширений и глубокая интеграция с современными инструментами делает его надёжным решением для профессионалов в крупных проектах. К тому же, гибкие настройки интерфейса и горячих клавиш позволяют адаптировать среду под индивидуальные потребности каждого программиста, обеспечивая комфорт и эффективность на всех этапах разработки.

# Системные требования

## Аппаратное обеспечение

- Оперативная память – от 1 ГБ
- Процессор – любой 1.6 ГГц и выше
- Жесткий диск – 500 МБ свободного места
- Монитор
- Клавиатура
- Мышь
- Интернет-соединение

## Программное обеспечение

- Операционная система
- Windows (от Windows 10 версия 1809)
- macOS (от macOS 10.15 Catalina)
- Linux (дистрибутивы, выпущенные с 2019-2020 годов и позже)

# Создание проекта

Для того, чтобы создать новый проект в Visual Studio Code достаточно создать новую папку у себя на устройстве. Шаги:

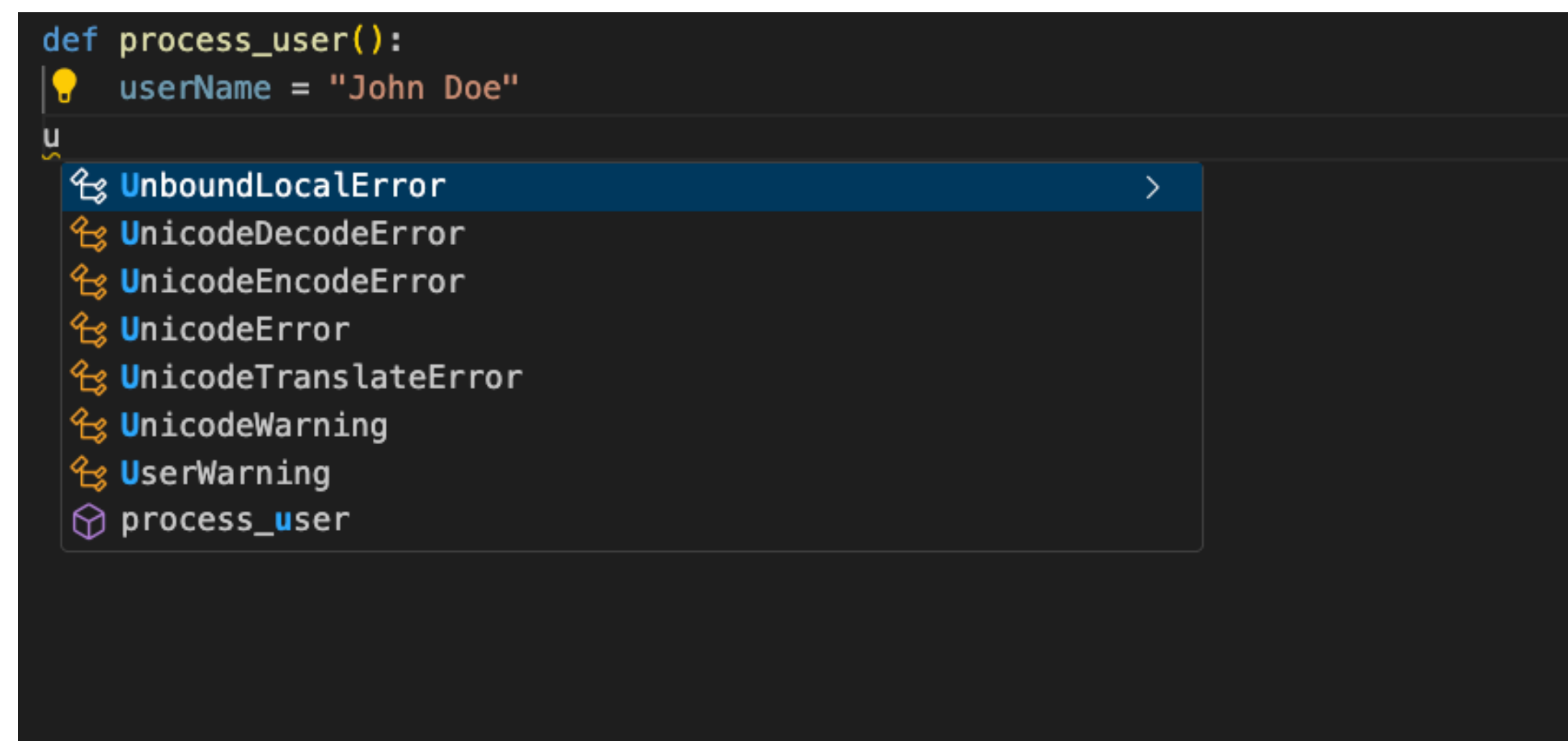
1. Создаем новую папку и переименовываем её
2. Нажимаем на созданную папку правой кнопкой мыши
3. Выбираем в появившемся меню «Открыть с помощью Code»

После этого папка откроется в Visual Studio Code и в ней можно добавлять новые файлы, писать в них код и т. д.

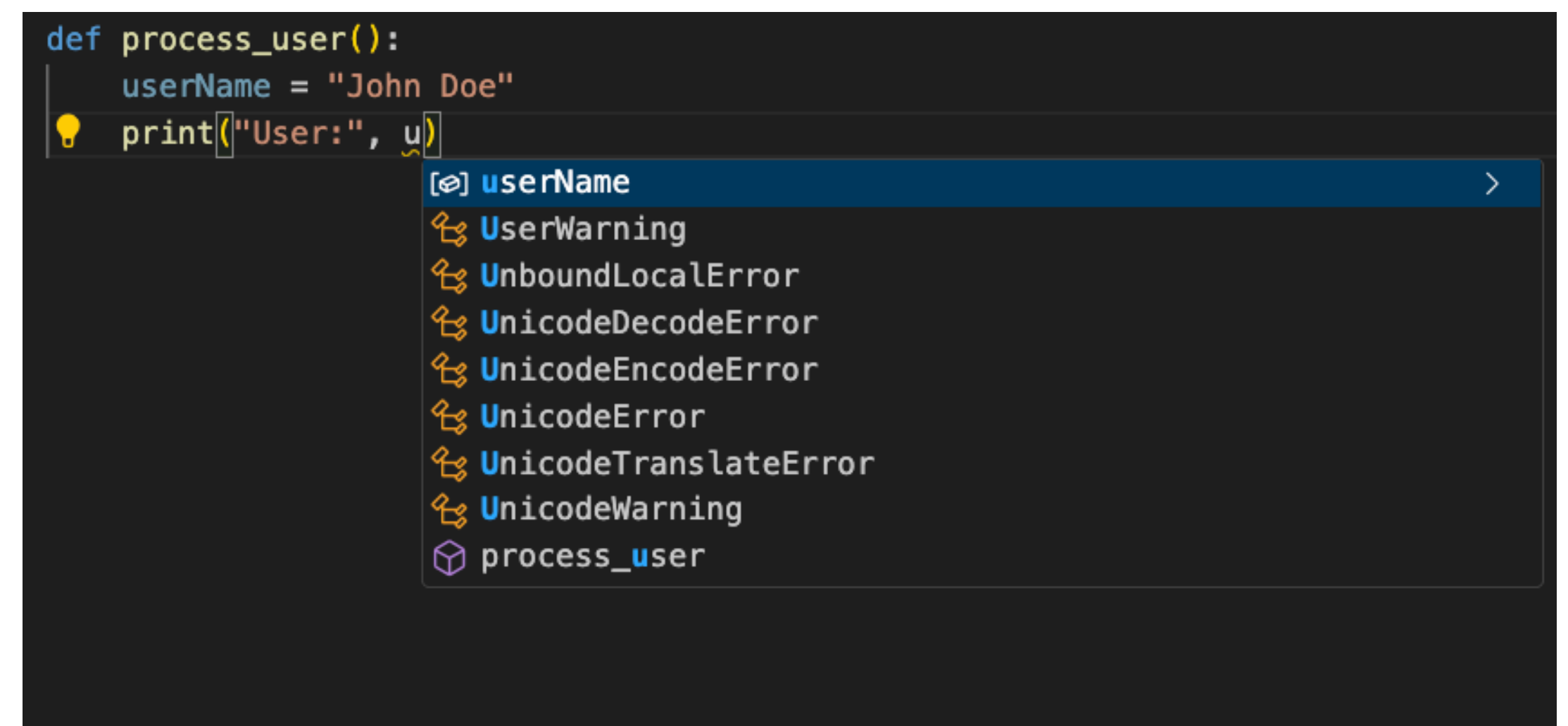
# IntelliSense: умная система подсказок

IntelliSense – одна из особенностей, которая серьёзно выделяет VS Code на фоне остальных редакторов. Например, если объявить переменную `userName` и начать вводить первые буквы, то среди прочего VS Code предложит и `userName`. Однако сделает он это не в любом месте, а там где это нужно по контексту: как только разработчик переходит к месту, где эта переменная становится нужна, IntelliSense начинает корректно предлагать `userName` в списке.

```
def process_user():  
    userName = "John Doe"  
    u
```



```
def process_user():  
    userName = "John Doe"  
    print("User:", u)
```



Пример разных ситуаций: в 1 случае вводить `userName` бессмысленно, поэтому IntelliSense предлагает более актуальные подсказки. А во 2 случае скорее всего программист захочет вывести переменную, и IntelliSense, понимая это, предлагает именно этот уже нужный вариант

# IntelliSense: умная система подсказок

Но при работе с такими низкоуровневыми языками как C/C++ (даже при установленных расширениях) IntelliSense даёт стандартные подсказки, зачастую не вникая в контекст даже правильного кода:

```
C example.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char userName[20] = "Hello";
6
7      printf("%s", user);
8
9      return 0;
10 }
```

- user\_addr\_t typedef u\_int64\_t user\_addr\_t
- user\_long\_t
- user\_off\_t
- user\_size\_t
- user\_ssize\_t
- user\_time\_t
- user\_ulong\_t
- [x] userName
- USER\_ADDR\_NULL
- \_\_USER\_LABEL\_PREFIX\_\_
- USE\_CLANG\_STDARG
- CAST\_USER\_ADDR\_T



# Форматирование кода: чистота и единый стиль

Автоматическое форматирование — встроенная функция VS Code, которая приводит код к единому, читаемому стилю одной горячей клавишей `Cmd+Shift+P`. Такие языки программирования как например js можно отформатировать без каких-либо проблем, но для редоктирования Python необходимо установить одно из специальных дополнений: `Autoper8`, `Black` или `Python Extension Pack`.

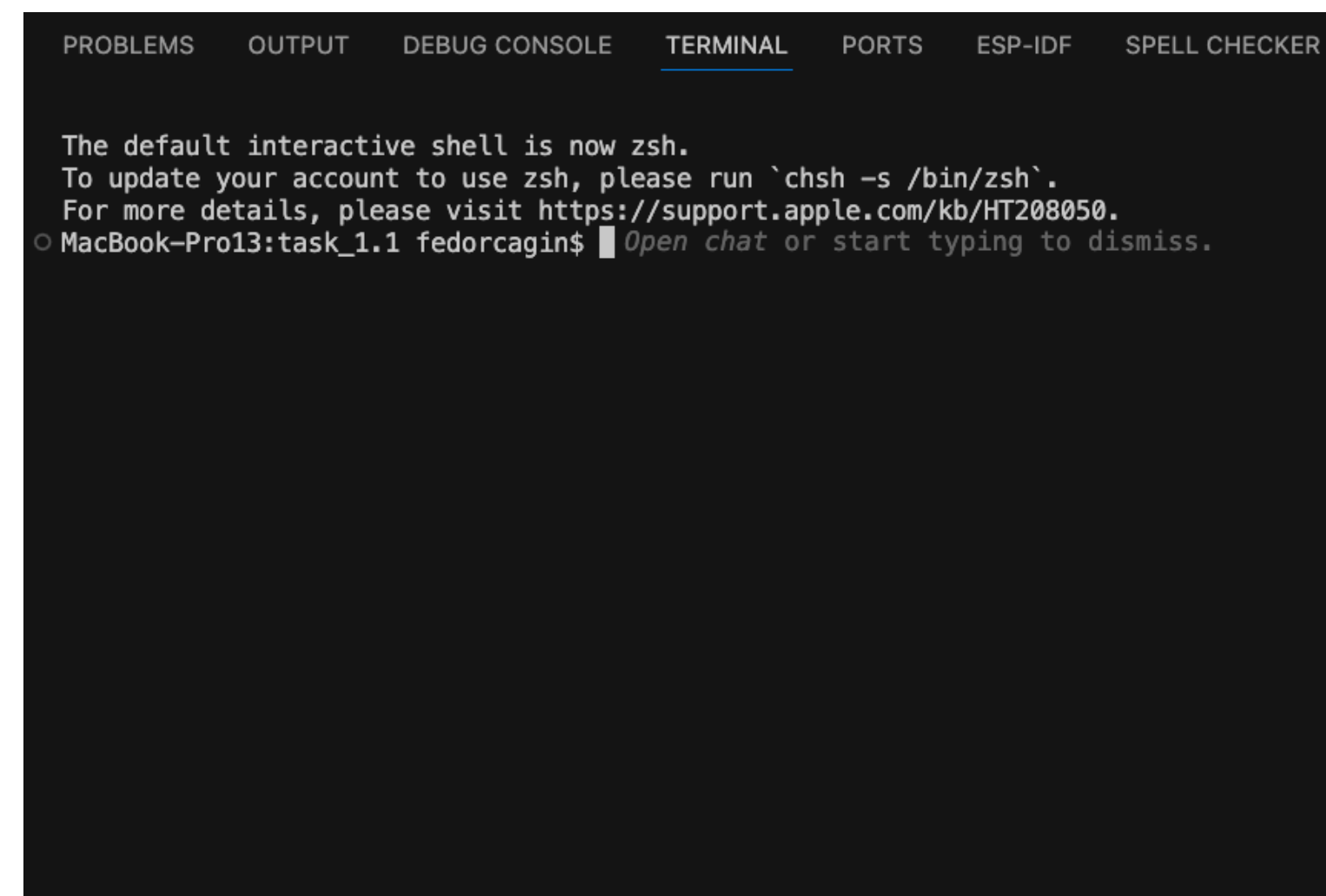
```
def process_user():  
    userName = "John Doe"  
    print("User:", userName)  
  
user_email = "john@example.com"  
    print("Email:", user_email)  
  
process_user()
```

`Cmd+Shift+P`  
→

```
def process_user():  
    userName = "John Doe"  
    print("User:", userName)  
  
user_email = "john@example.com"  
  
print("Email:", user_email)  
  
process_user()
```

# Интегрированный терминал: всё в одном окне

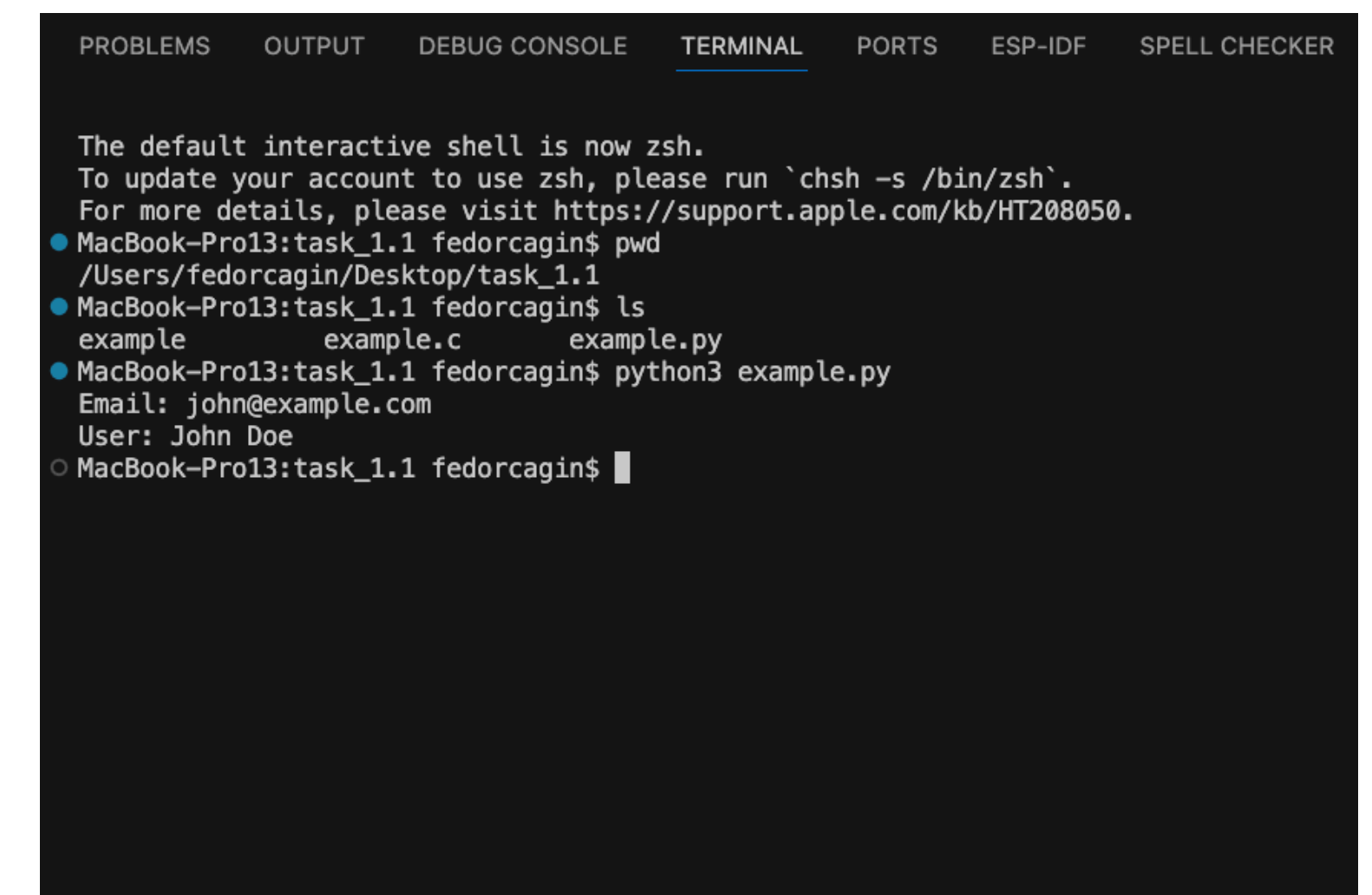
Интегрированный терминал (Ctrl + `) позволяет заниматься запуском и отладкой кода без переключения между приложениями. С помощью него можно работать с командной строкой прямо в интерфейсе VS Code. Он позволяет выполнять команды непосредственно в среде разработки.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF SPELL CHECKER

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro13:task_1.1 fedorcagin$ █ Open chat or start typing to dismiss.
```

На примере видно, как без переключения окон можно проверить содержимое папки (ls) и запустить программу (python3 example.py), сразу увидев результат её работы.




```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF SPELL CHECKER

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro13:task_1.1 fedorcagin$ pwd
/Users/fedorcagin/Desktop/task_1.1
MacBook-Pro13:task_1.1 fedorcagin$ ls
example      example.c    example.py
MacBook-Pro13:task_1.1 fedorcagin$ python3 example.py
Email: john@example.com
User: John Doe
MacBook-Pro13:task_1.1 fedorcagin$ █
```



# Дополнения


К тому же, приложение может похвастаться огромной библиотекой доступных дополнений: магазин содержит десятки тысяч плагинов для различных языков программирования, фреймворков и инструментов. Это позволяет адаптировать редактор под конкретные нужды проекта, добавляя нужные возможности, что превращает VS Code из простого редактора в мощную среду разработки.




ESP-IDF

Espressif Systems [espressif.com](#) | 1,121,472 | ★★★★★ (128)

Develop and debug applications for Espressif chips with ESP-IDF


[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ☒ Auto Update 




C/C++ pre-release

Microsoft [microsoft.com](#) | 87,999,216 | ★★★★★ (589)

C/C++ IntelliSense, debugging, and code browsing.


[Disable](#) [Uninstall](#) ☒ Auto Update 




Blender Python Code Templates

blenderfreetimeprojects | 52,557 | ☆☆☆☆☆

Templates for Blender Addon development. Templates translated are from Blender


[Disable](#) [Uninstall](#) ☒ Auto Update 




Julia

julialang [julia-vscode.org](#) | 941,342 | ★★★★★ (82)

Julia Language Support


[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ☒ Auto Update 

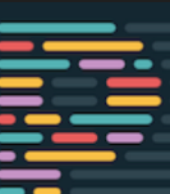


Live Server


Ritwick Dey | 68,780,341 | ★★★★★ (511)

Launch a development local Server with live reload feature for static & dynamic


[Install](#) ☒ Auto Update 




Prettier - Code formatter

Prettier [prettier.io](#) | 61,193,037 | ★★★★★ (486) |  Sponsor

Code formatter using prettier


[Install](#) ☒ Auto Update 

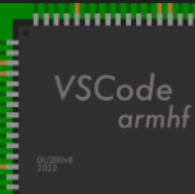


GitLens — Git supercharged

GitKraken [gitkraken.com](#) | 44,007,492 | ★★★★★ (876)

Supercharge Git within VS Code — Visualize code authorship at a glance via Git


[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ☒ Auto Update 




Arm Assembly

dan-c-underwood | 909,510 | ★★★★★ (7)

Arm assembly syntax support for Visual Studio Code


[Disable](#) [Uninstall](#) ☒ Auto Update 




Docker

Microsoft [microsoft.com](#) | 47,533,426 | ★★★★★ (105)

Makes it easy to create, manage, and debug containerized applications.


[Disable](#) [Uninstall](#) ☒ Auto Update 




AREPL for python


Almenon | 1,744,713 | ★★★★★ (107)

real-time python scratchpad


[Disable](#) [Uninstall](#) ☒ Auto Update 




Material Icon Theme

Philipp Kief | 30,890,917 | ★★★★★ (378) |  Sponsor

Material Design Icons for Visual Studio Code


[Install](#) ☒ Auto Update 



Auto Rename Tag

Jun Han | 23,104,654 | ★★★★★ (207)

Auto rename paired HTML/XML tag

[Install](#) ☒ Auto Update 

# Интеграция с системой контроля версий Git

В Visual Studio Code встроена поддержка Git, что делает работу более удобной.

Основные возможности:

- Создание Git репозитория
- Отслеживание изменений
- Коммиты
- Ветвление и слияние
- Просмотр истории



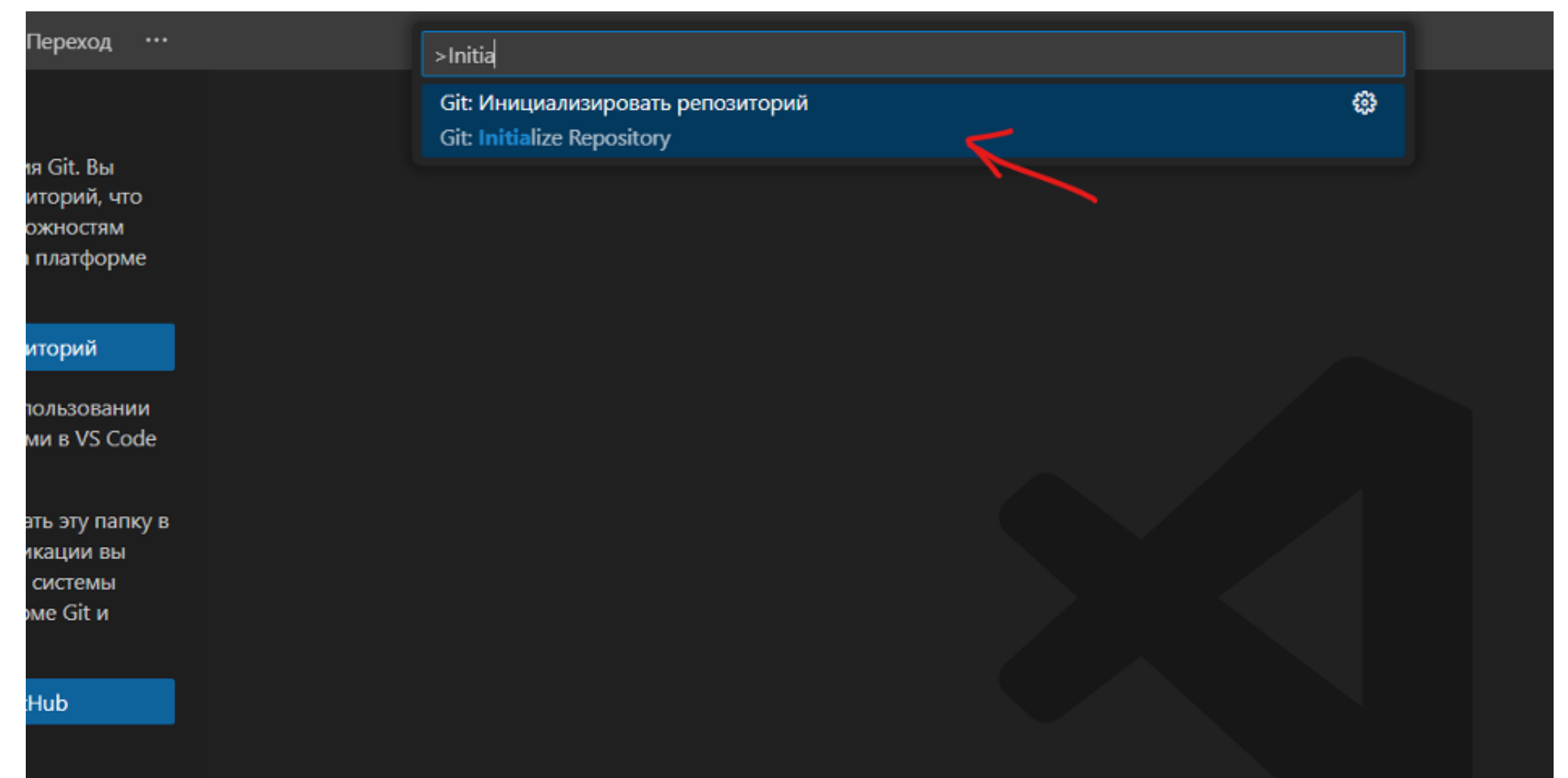
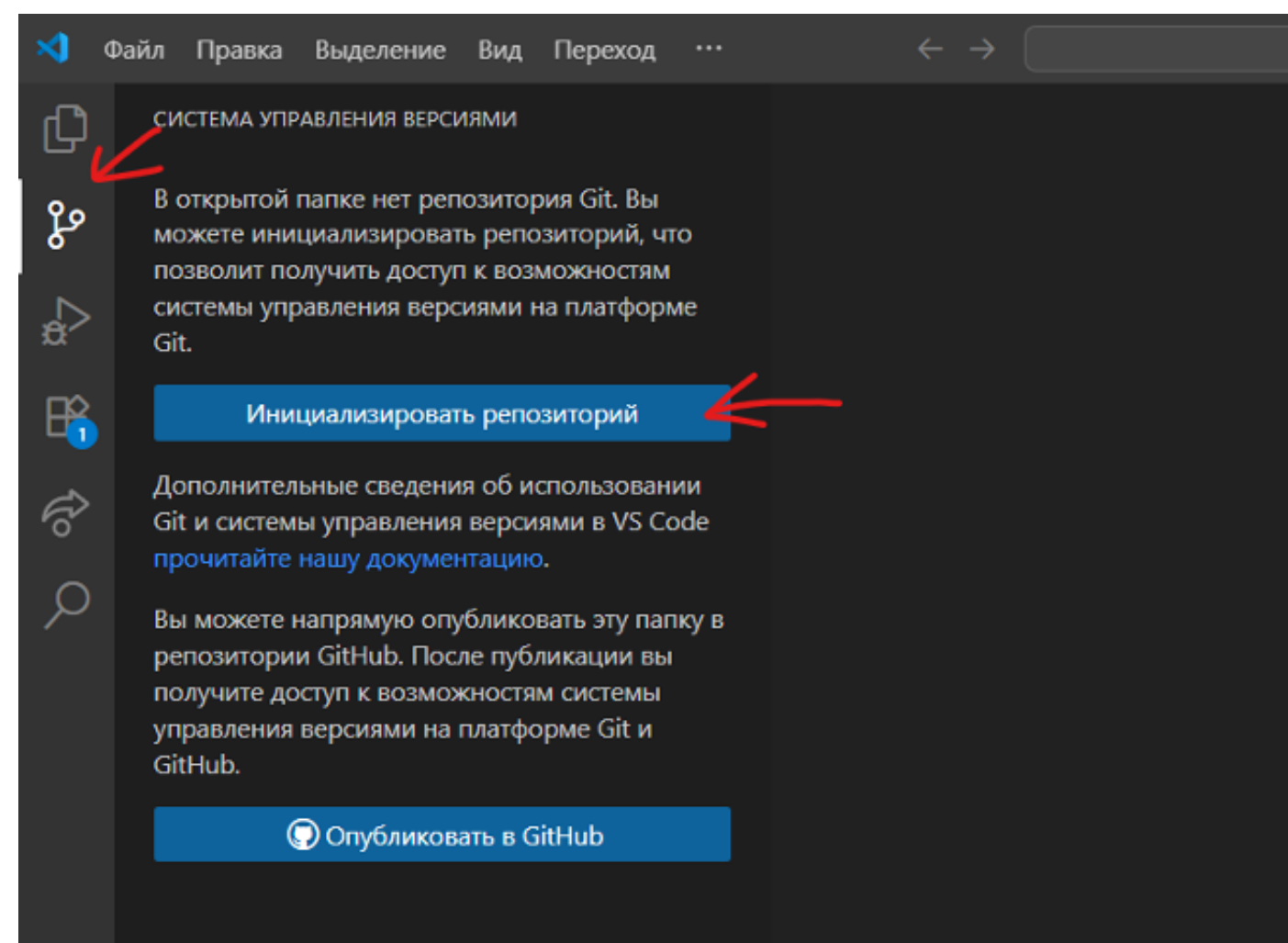
# Интеграция с системой контроля версий Git

## Как создать репозиторий

Прямо из VS Code. Чтобы это сделать перейдите во вкладку «Система управления версиями» и нажмите кнопку «Инициализировать репозиторий».

Или

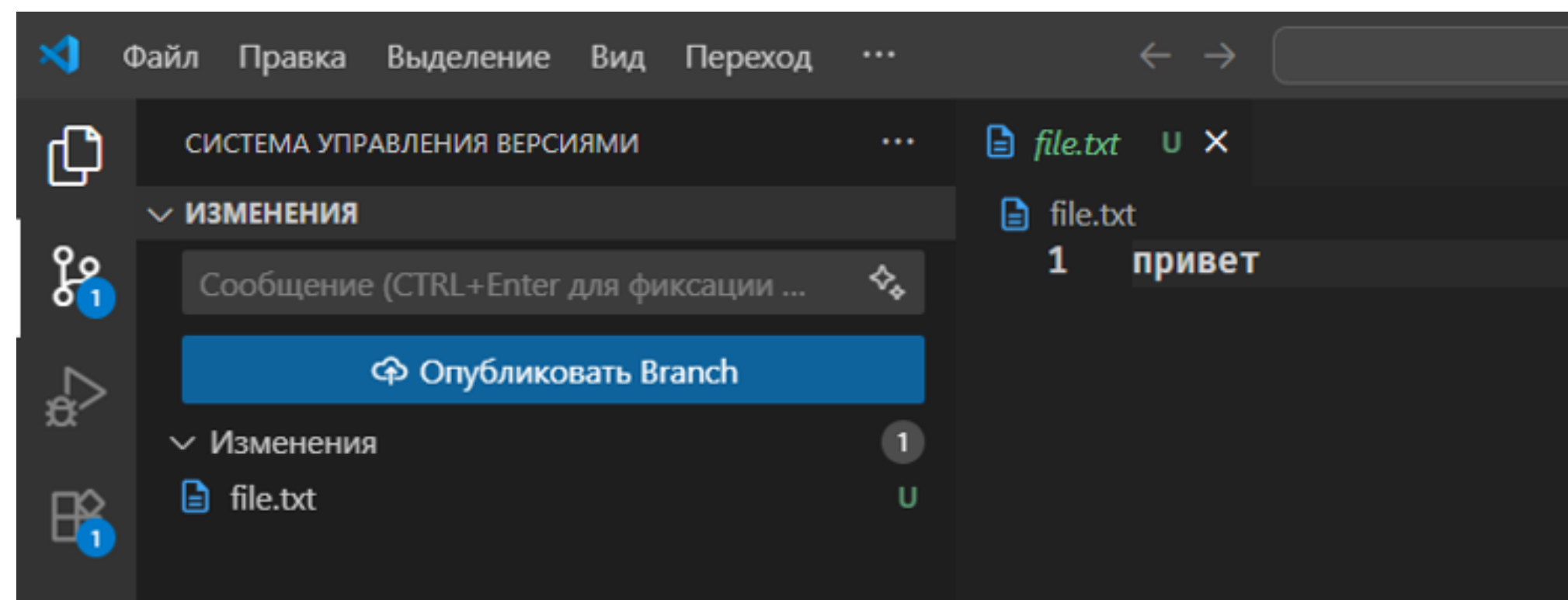
Нажать сочетание клавиш Ctrl+Shift+P, затем найти и выбрать команду «Git: Инициализировать репозиторий» и нажать клавишу Enter.



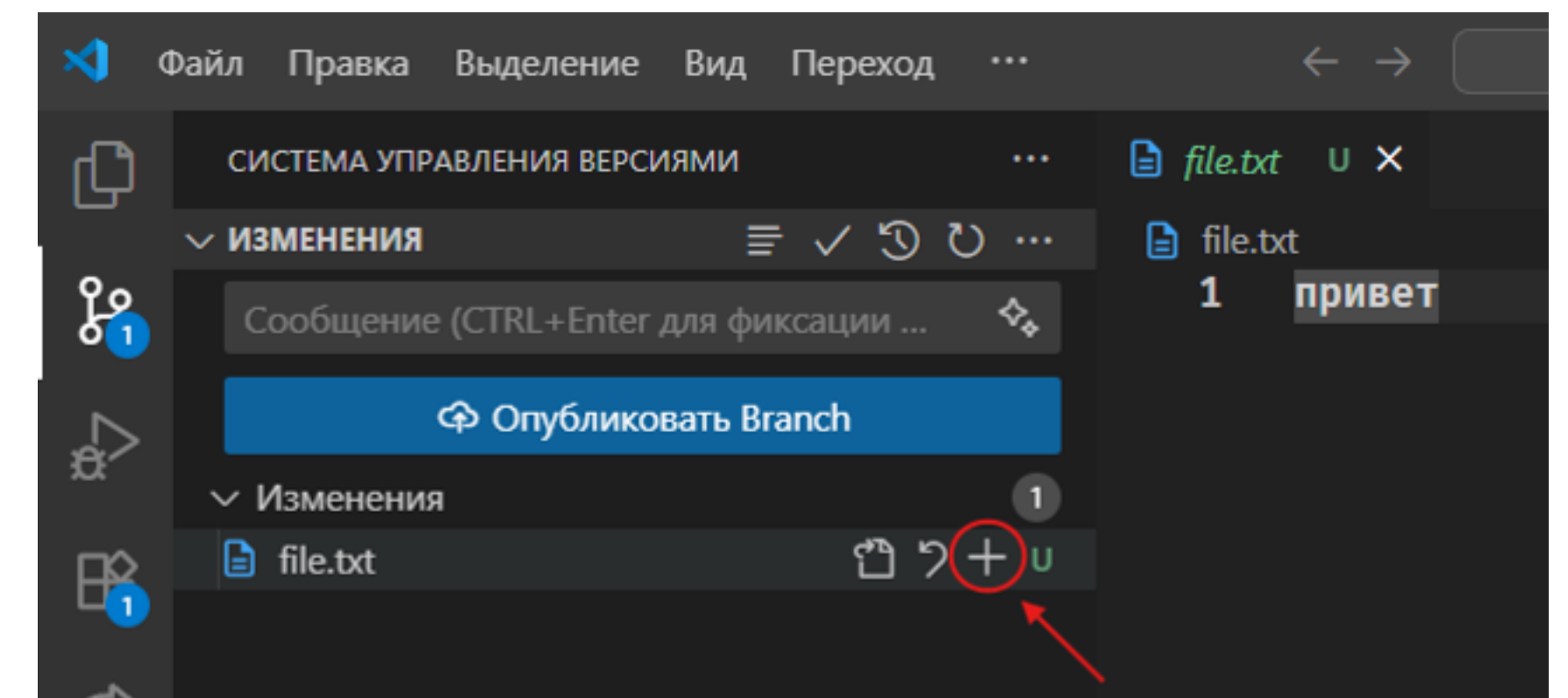


# Интеграция с Git: отслеживание изменений

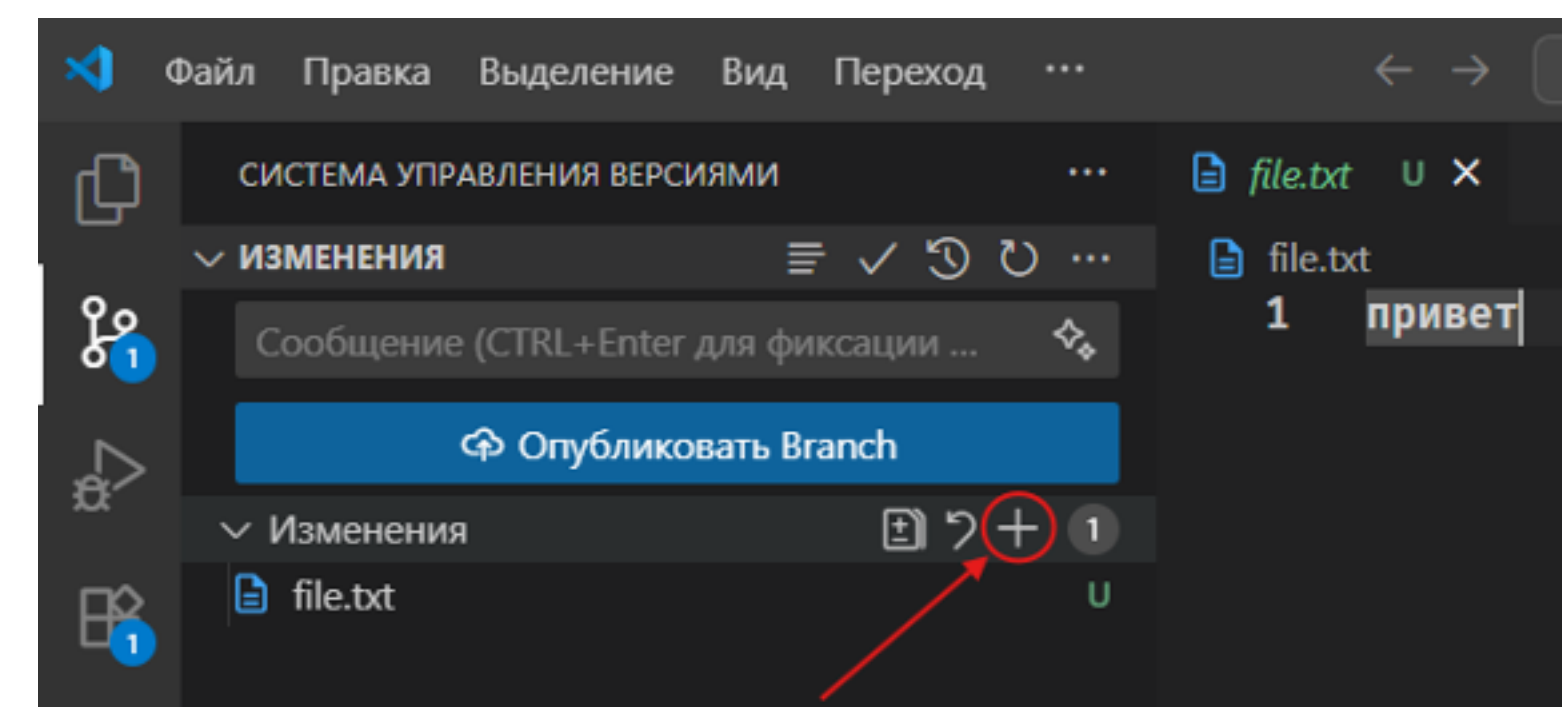
При инициализированном репозитории, в этой вкладке будут отображаться все измененные файлы.



Измененные файлы можно выборочно добавлять в индекс Git, для этого надо расположить курсор на файл и нажать на иконку плюса.

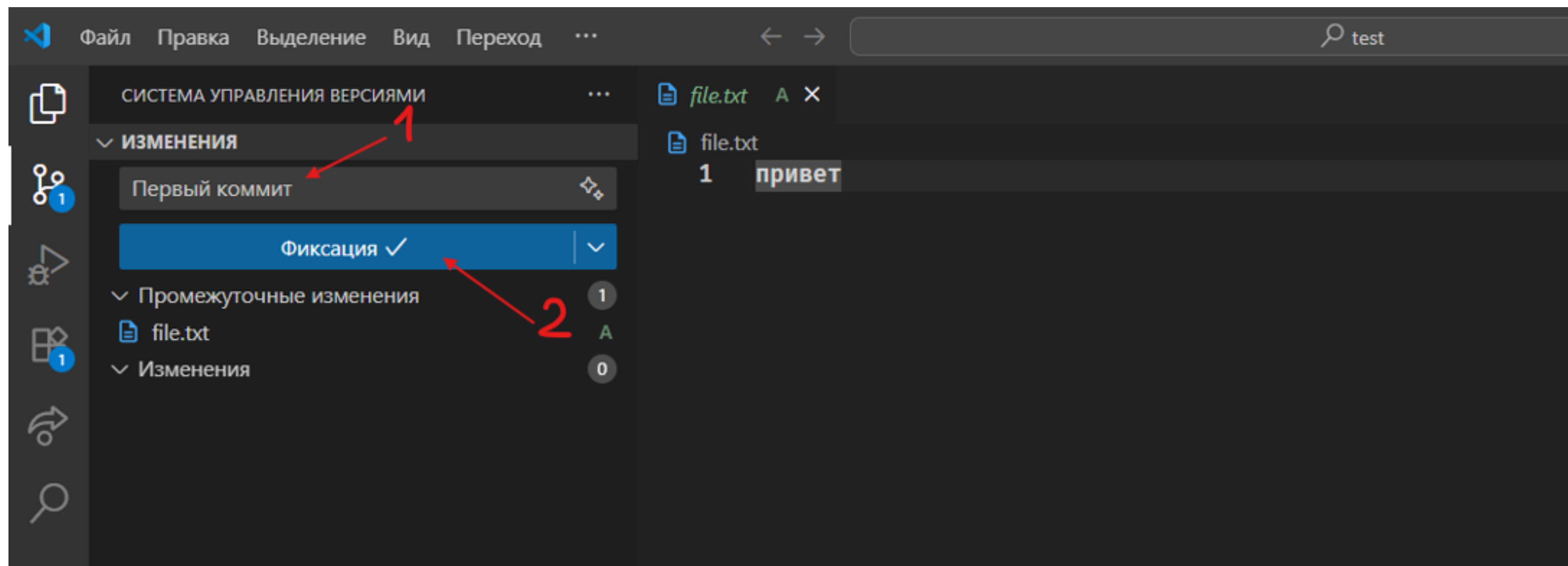


Или проиндексировать сразу все измененные файлы, наведясь на поле «Изменения» и нажав иконку плюса правее.



# Интеграция с Git: создание коммитов

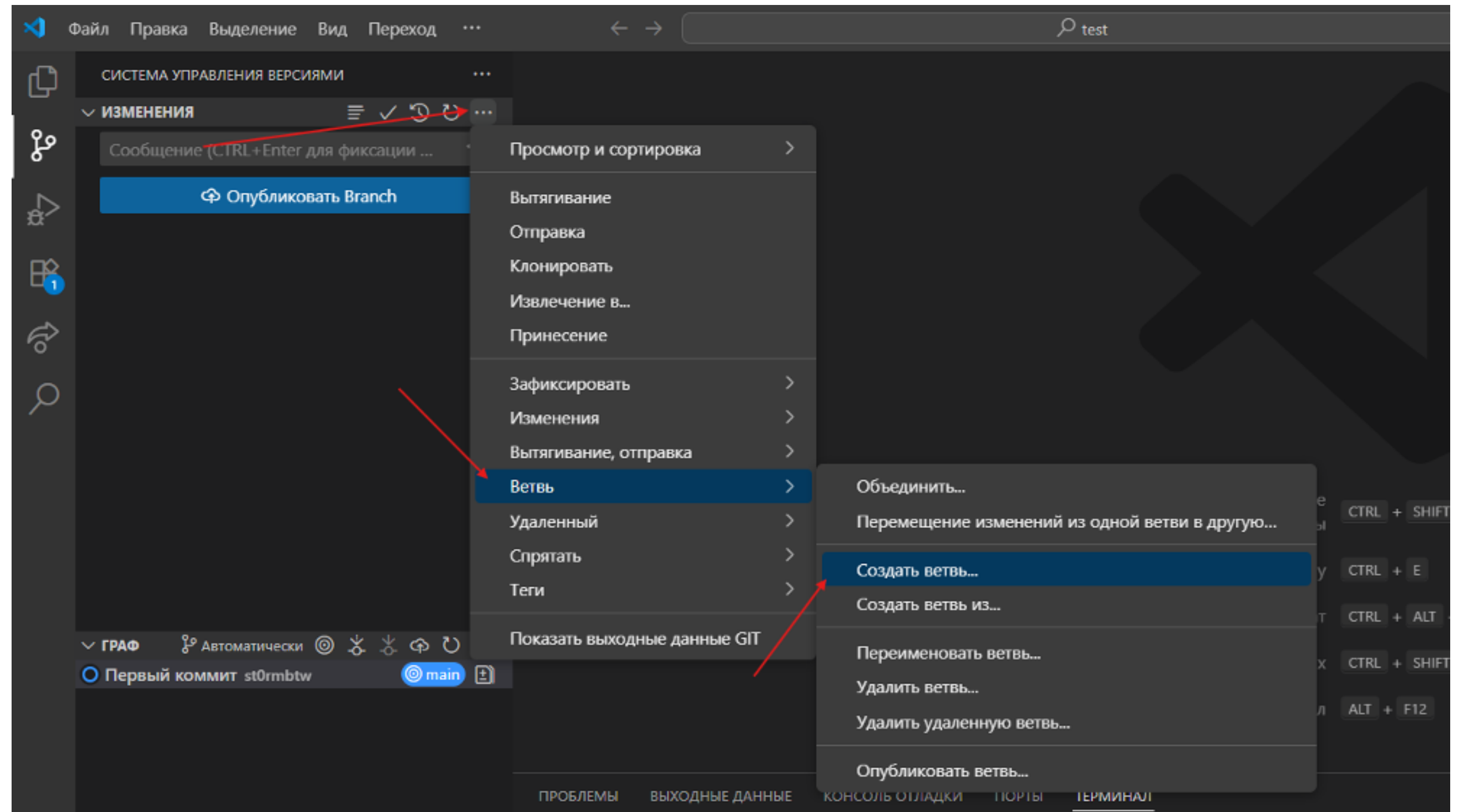
Когда все нужные изменения были проиндексированы, можно их закоммитить. Для этого необходимо ввести название коммита в текстовом поле сверху и нажать кнопку «Фиксация».



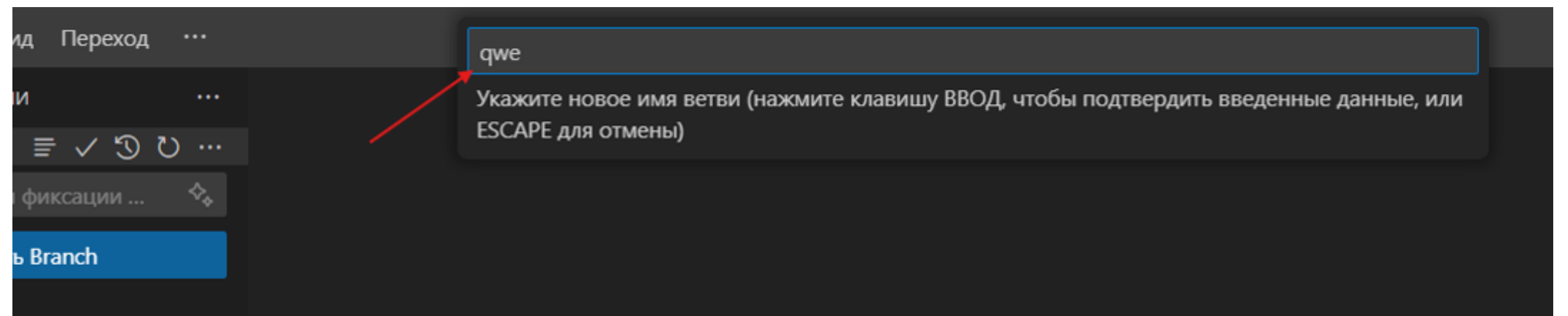


# Интеграция с Git: ветвление

Поддерживается создание новых веток. Чтобы создать новую ветку наведите курсором на заголовок «Изменения» и откройте контекстное меню, нажав на три точки правее. Затем нажмите на кнопку «Создать ветвь» из пункта «Ветвь».

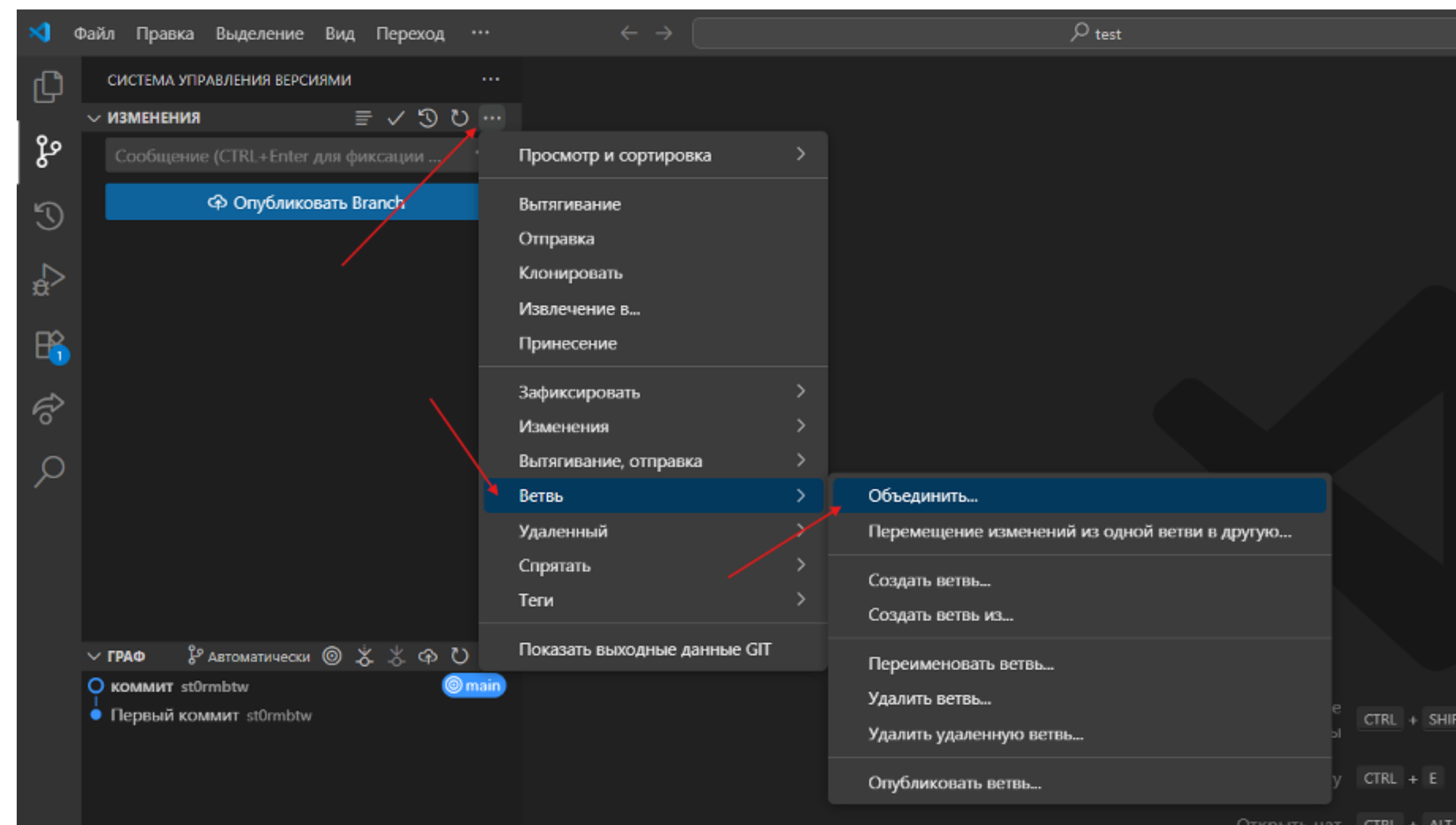


В появившемся сверху поле ввода текста введите название новой ветки и нажмите Enter.



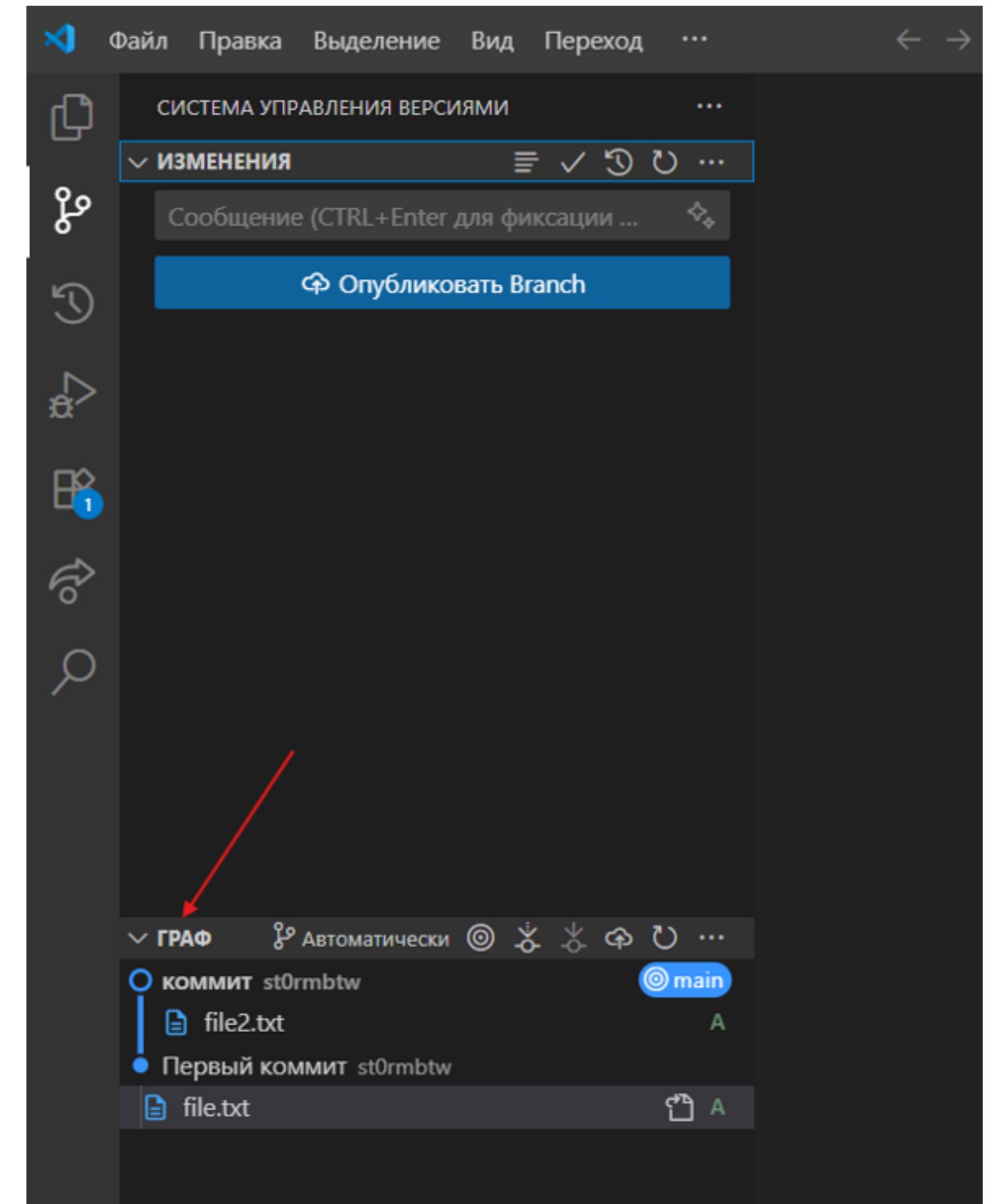
# Интеграция с Git: слияние

Чтобы слить ветки друг с другом, откройте контекстное меню и нажмите кнопку «Объединить» в пункте «Ветвь», и затем в появившемся списке выберите ту ветвь, из которой хотите слить изменения, находясь на ветке, в которую хотите слить изменения.



# Интеграция с Git: просмотр истории

Можно просматривать историю изменений. В меню системы управления версиях, под пунктом «Изменения» есть пункт «Граф». В нём отображается история всех коммитов, а также их содержимое на текущей ветке. Можно посмотреть изменения в каждом файле внутри коммита.



# AI Функции: GitHub Copilot

Ещё, в Visual Studio Code есть встроенный GitHub Copilot, нейросеть, которая помогает в программировании. К сожалению, протестировать её не получилось, при попытке зарегистрироваться происходит ошибка. Вероятно, ошибка связана с тем, что GitHub Copilot недоступен пользователям из России.

