

Инвариантное задание 1.1

Практика

Конспект Visual Studio Code

Общая характеристика

Visual Studio Code (VS Code) — это редактор кода для разных языков программирования. Он относительно немного весит, прост в освоении, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках. Редактор кода существует для всех популярных операционных систем: Windows, Linux и macOS. Он бесплатный, в отличие от большинства версий «старшего брата» Visual Studio.

VS Code позволяет легко писать, форматировать и редактировать код на разных языках. С его помощью можно быстро создать проект и структуру файлов в нем, он подсвечивает синтаксис кода и помогает автоматически править ошибки. В нем есть возможности для отладки и запуска кода на некоторых языках. Редактор легко расширяется, поэтому к перечисленным функциям можно добавить новые — достаточно просто скачать нужное дополнение из официального каталога. Дополнения тоже распространяются бесплатно.

Системные требования

Visual Studio Code имеет небольшой вес и легко запускается на современном оборудовании.(<500 МБ)

Рекомендуемые требования:

- 1,6 ГГц или более быстрый процессор
- 1 ГБ оперативной памяти

VS Code поддерживается на следующих платформах:

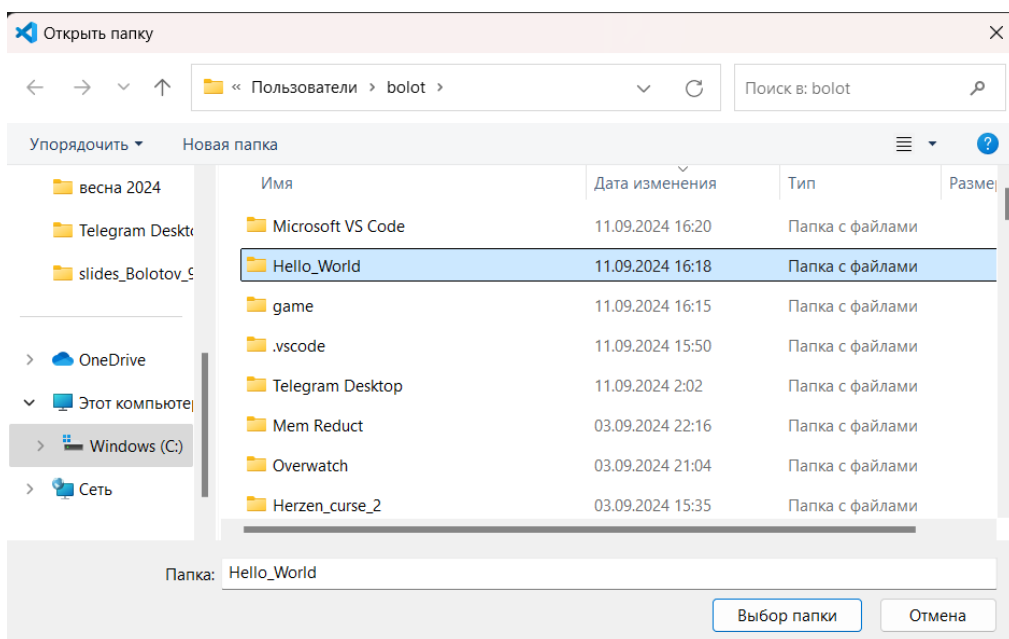
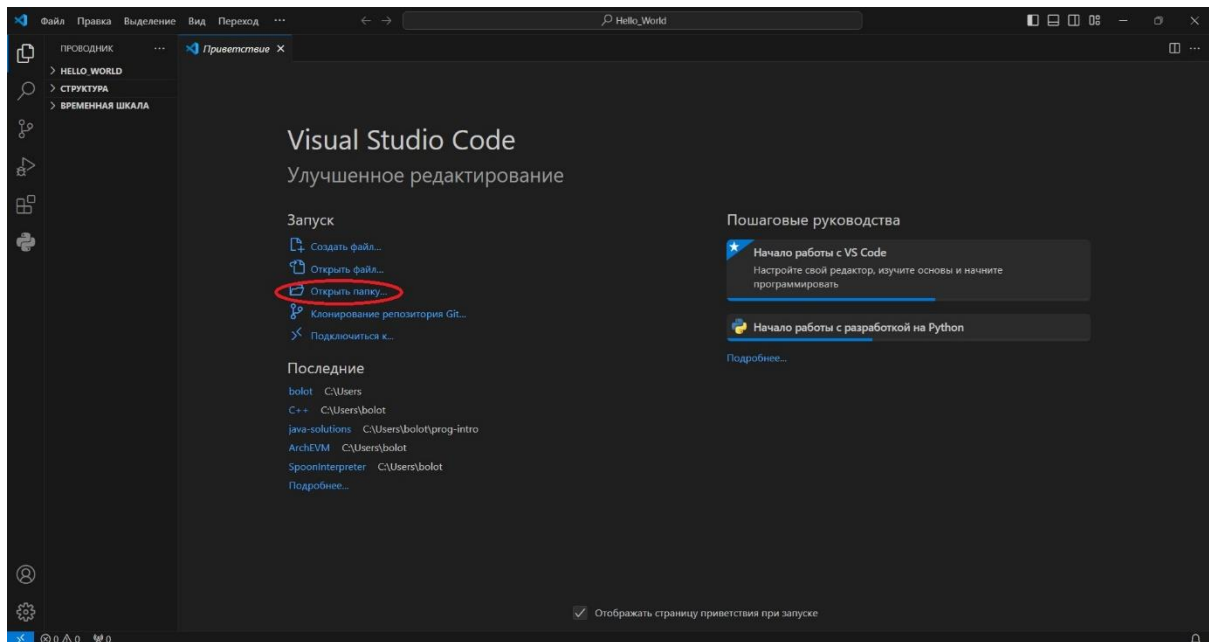
- Windows 10 и 11 (64-бит)
- версии macOS с поддержкой обновлений безопасности Apple. Как правило, это последний выпуск и две предыдущие версии.
- Linux (Debian): Ubuntu Desktop 20.04, Debian 10
- Linux (Red Hat): Red Hat Enterprise Linux 8, Fedora 36

В Visual Studio Code есть инструменты, чтобы писать, отлаживать и тестировать код. Функции VS Code можно расширять с помощью расширений, изменять интерфейс и упрощать работу с кодом.

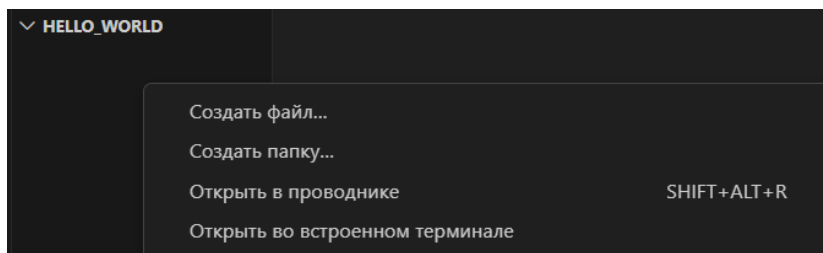
Функции

Создание проекта

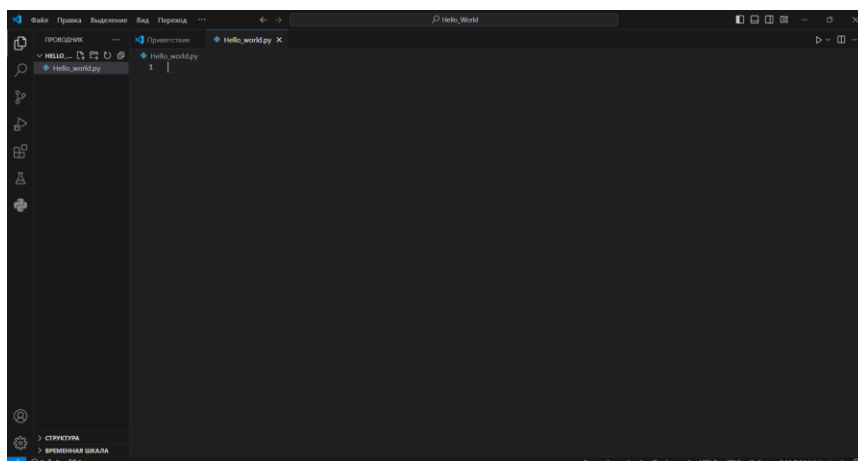
Для создания проекта создаём пустую папку и открываем её в VS Code.



Теперь в данной папке можно создавать нужные файлы и дополнительные папки для проекта

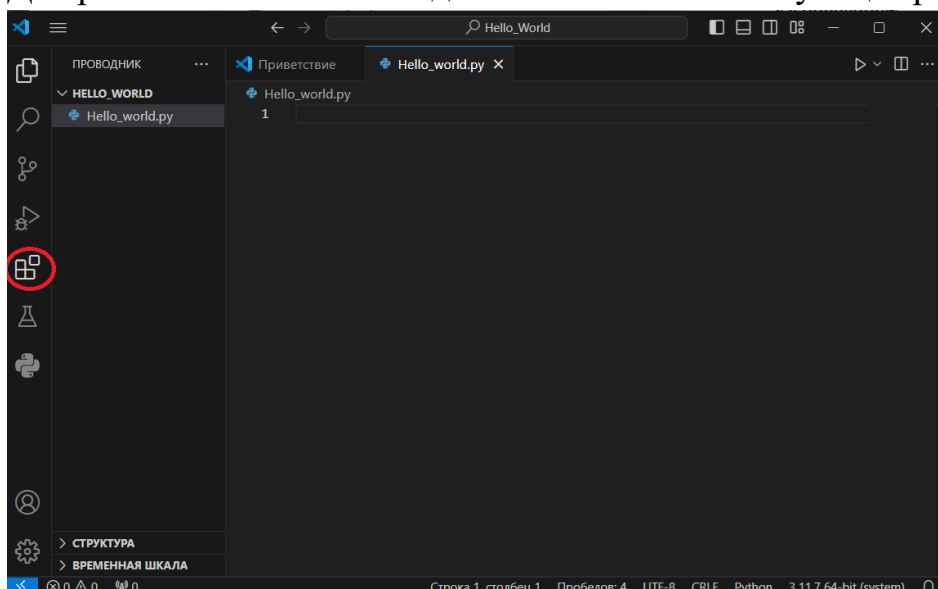


Создаём файл указывая его имя и расширение. Теперь в этом файле можно работать.



Кодирование

В VS Code это среда для работы с разными языками программирования. Для работы с ними необходимо найти соответствующее расширение.





Допустим расширение для работы с Python. Также есть расширение IntelliCode — интеллектуальный ассистент кодирования, который делает автодополнение кода, проверяет синтаксис, сообщает об ошибках и даёт рекомендации по их исправлению.

Форматирование кода

Используйте горячие клавиши: в VS Code нажмите Alt+Shift+F (для Windows/Linux) или Option+Shift+F (для macOS), чтобы применить ручное форматирование кода.

```

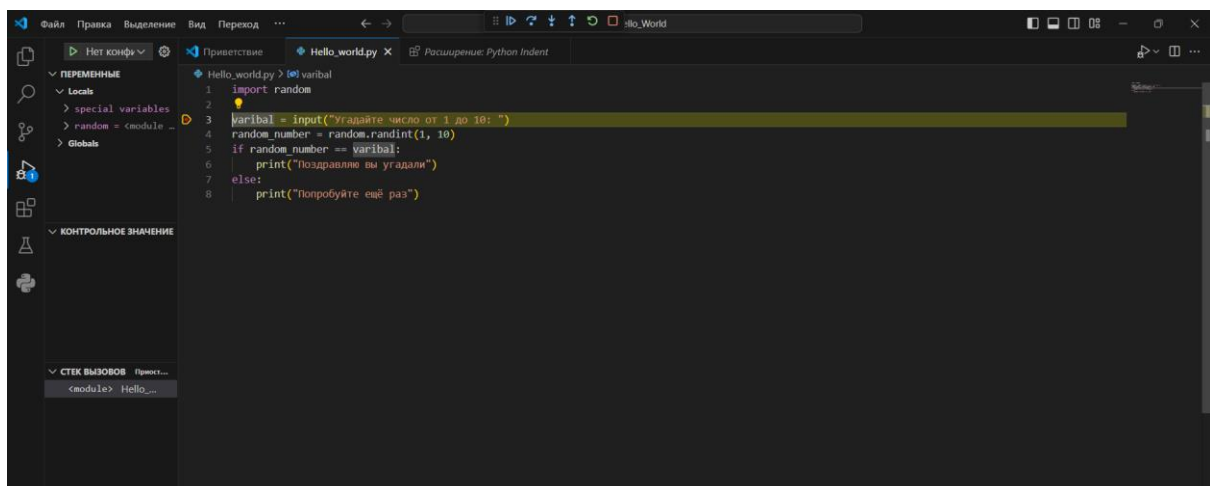
Hello_world.py > ...
1  import random
2
3  varibal = input("Угадайте число от 1 до 10")
4  random_number = random.randint(1, 10)
5  if random_number == varibal:
6  print("Поздравляю вы угадали")
7  else:
8  print("Попробуйте ещё раз")
```

```
Hello_world.py > ...
1  import random
2
3  varibal = input("Угадайте число от 1 до 10")
4  random_number = random.randint(1, 10)
5  if random_number == varibal:
6      print("Поздравляю вы угадали")
7  else:
8      print("Попробуйте ещё раз")
```

Отладка

Есть два способа начать отладку:

1. Нажать Ctrl + Shift + D
2. Щелкнуть на вкладку Debug.



Запуск

Есть два способа запуска программы:

1. Использование клавиши F5.
2. Нажать на кнопку Run code.

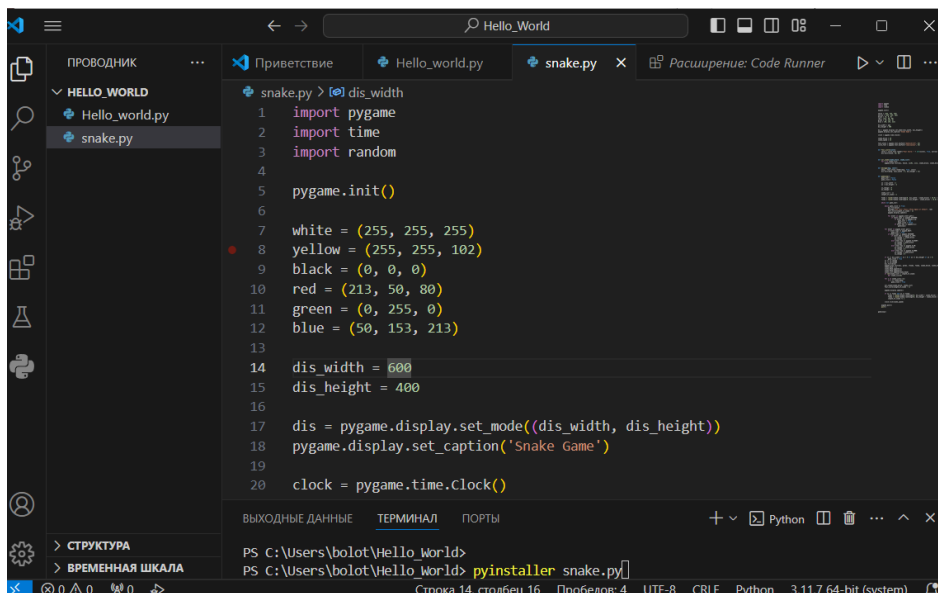
```
Hello_world.py > ...
1  import random
2
3  varibal = input("Угадайте число от 1 до 10: ")
4  random_number = random.randint(1, 10)
5  if random_number == varibal:
6      print("Поздравляю вы угадали")
7  else:
8      print("Попробуйте ещё раз")
```

ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ ПОРТЫ

```
PS C:\Users\bolot\Hello_World> & C:/msys64/ucrt64/bin/python.exe c:/Users/bolot/Hello_World/Hello_world.py
Угадайте число от 1 до 10: 1
Попробуйте ещё раз
PS C:\Users\bolot\Hello_World> & C:/msys64/ucrt64/bin/python.exe c:/Users/bolot/Hello_World/Hello_world.py
Угадайте число от 1 до 10: 
```

Компиляция

Компиляцию можно провести с помощью модуля Pyinstaller. Для этого установим с помощью команды ‘pip install pyinstaller’, введённой в Terminal. Потом переходим в нужную папку и вводим ‘pyinstaller название_проекта.py’, также в Terminal.



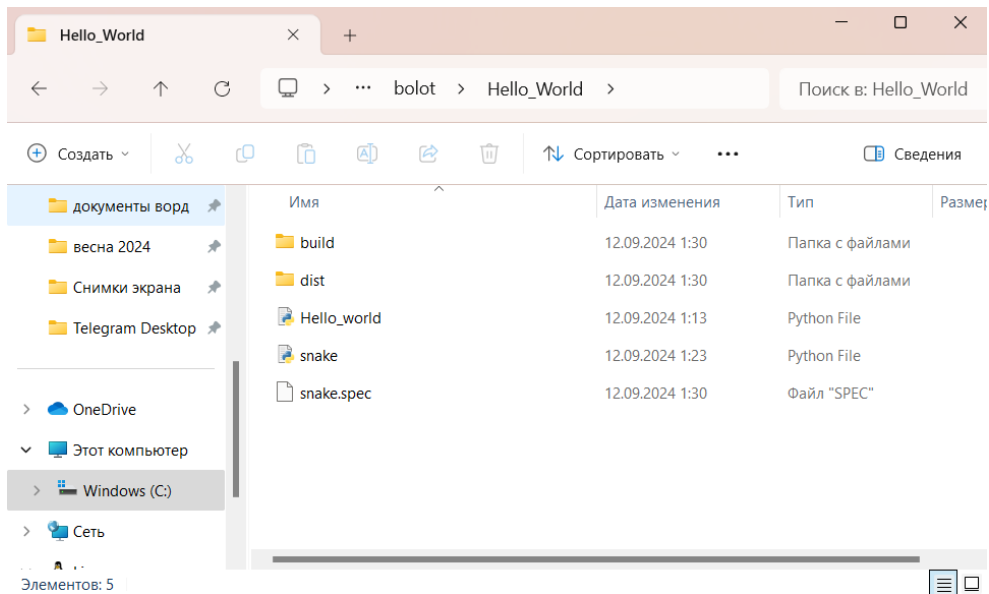
```
snake.py > dis_width
1  import pygame
2  import time
3  import random
4
5  pygame.init()
6
7  white = (255, 255, 255)
8  yellow = (255, 255, 102)
9  black = (0, 0, 0)
10 red = (213, 50, 80)
11 green = (0, 255, 0)
12 blue = (50, 153, 213)
13
14 dis_width = 600
15 dis_height = 400
16
17 dis = pygame.display.set_mode((dis_width, dis_height))
18 pygame.display.set_caption('Snake Game')
19
20 clock = pygame.time.Clock()
```

ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ ПОРТЫ

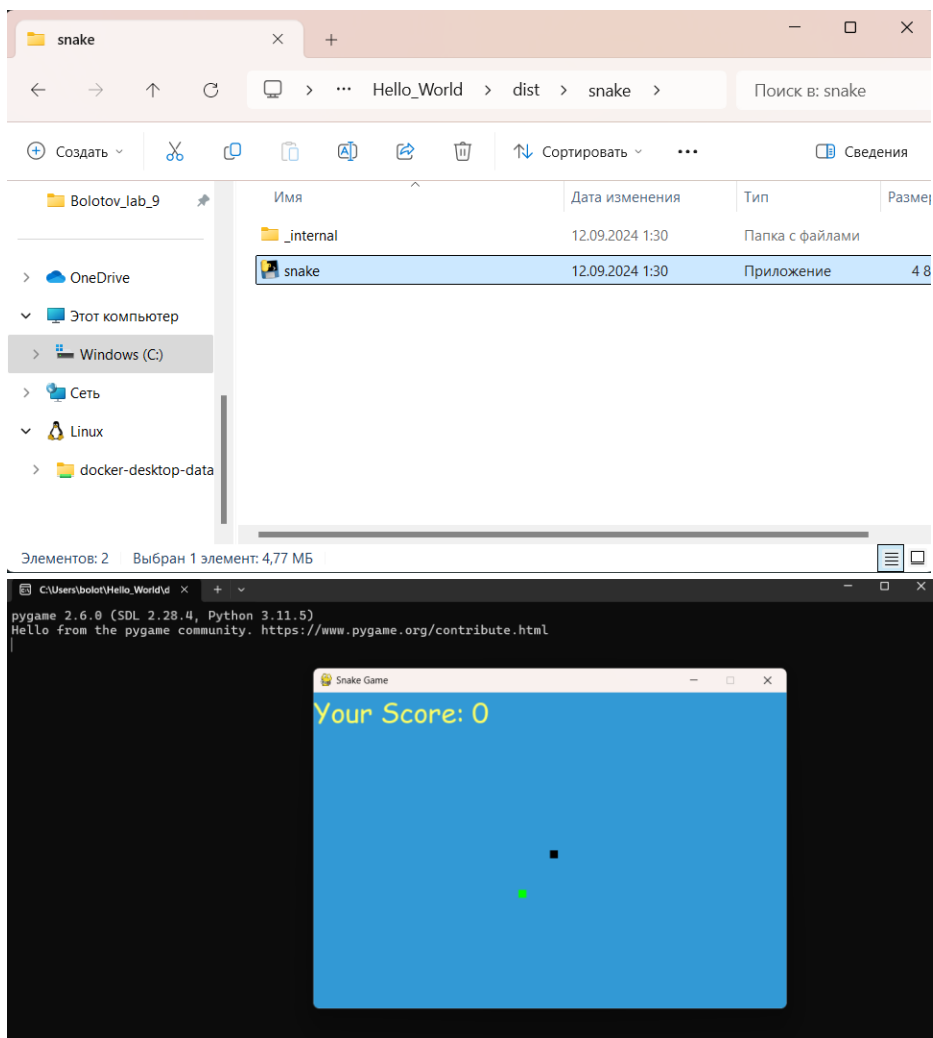
```
PS C:\Users\bolot\Hello_World>
PS C:\Users\bolot\Hello_World> pyinstaller snake.py
```

Строка 14, столбец 16 Пробелов: 4 UTF-8 CRLF Python 3.11.7 64-bit (system)

После нажатия Enter в папке с проектом появится несколько новых папок, в том числе dist, внутри которой и будет файл с программой формата .exe.



Его можно запустить:

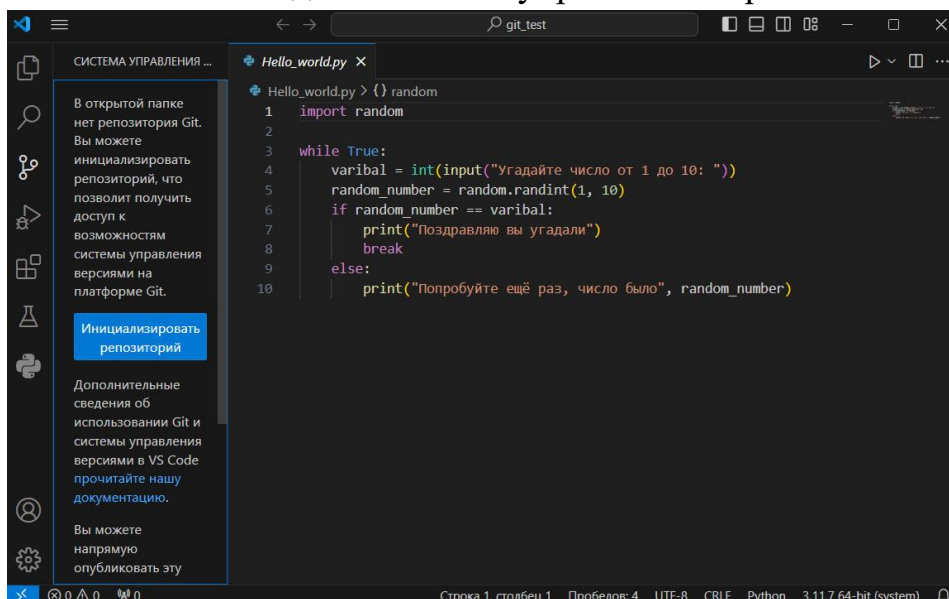


Версионирование

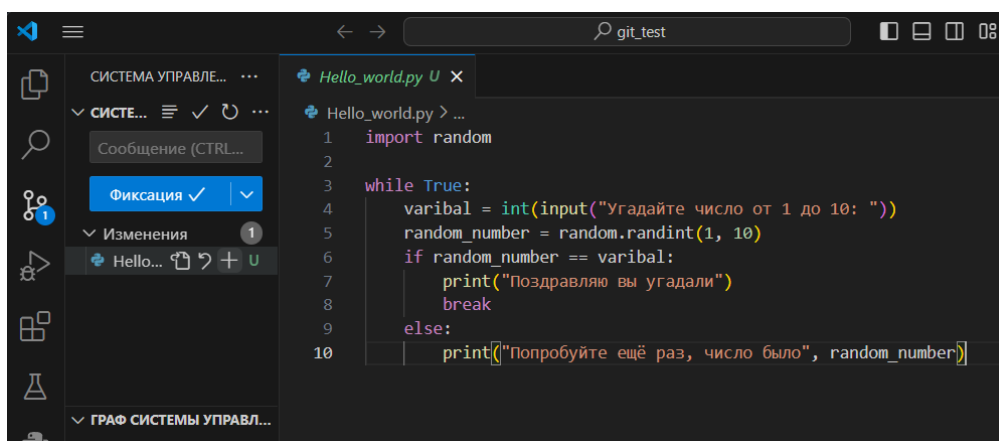
Visual Studio Code поддерживает работу с системами контроля версий GitHub. Например для упрощения существует «Semantic versioning» — это способ автоматического версионирования проекта в зависимости от описания commit-message. Для того, чтобы проделать такое версионирование задействуется менеджер автоверсионирования проекта semantic-release, который использует знания о типе совершённых commit-ов исходя из commit-message для определения степени влияния внесенных изменений в проект.

Публикация в репозитории на GitHub

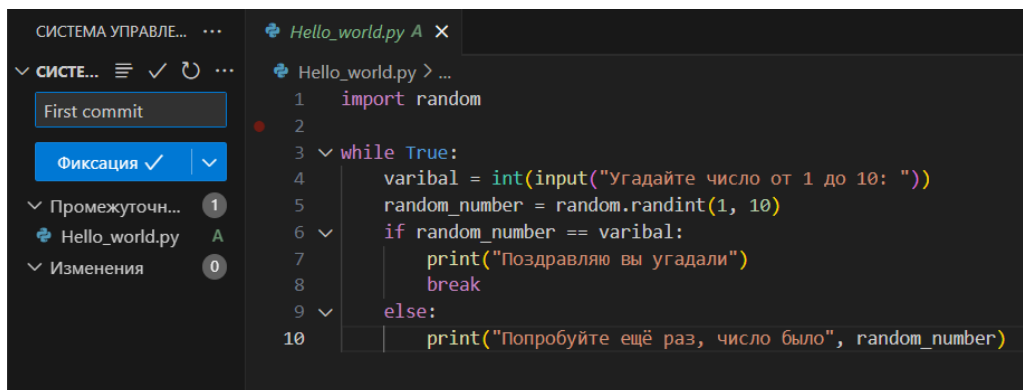
Чтобы пользоваться GitHub нужно перейти в папку и инициализировать там GitHub в вкладке «Система управления версиями»



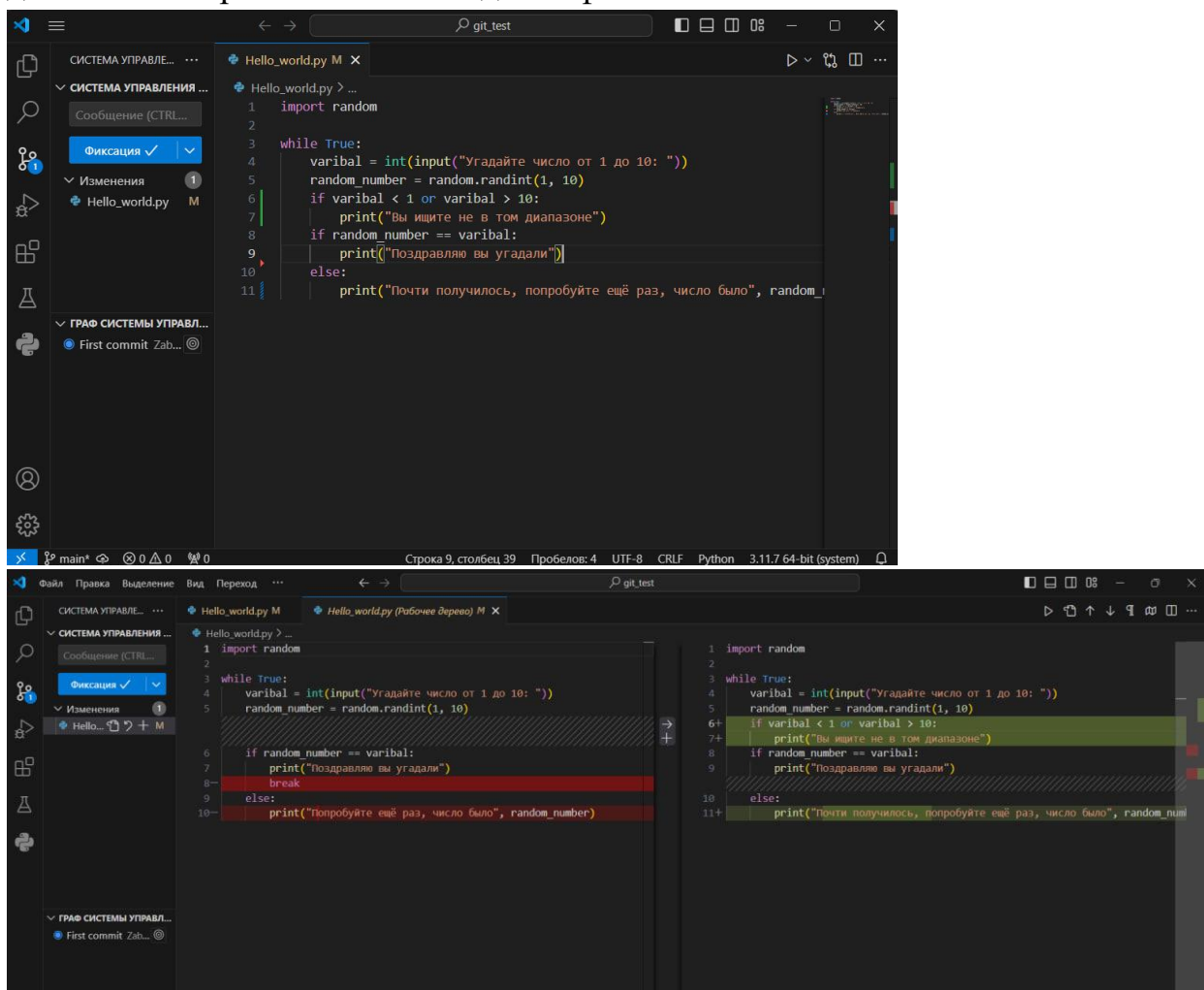
Следующий шаг — нажать на плюсик на «Unversioned» файлах.



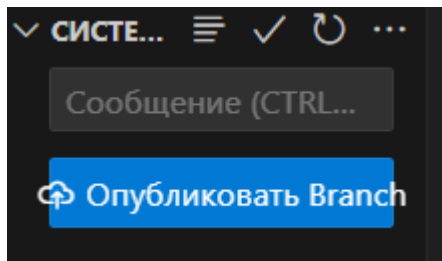
Они переходят в состояние «Added», остаётся только написать сообщение к данному Commit и он будет добавлен в репозиторий.



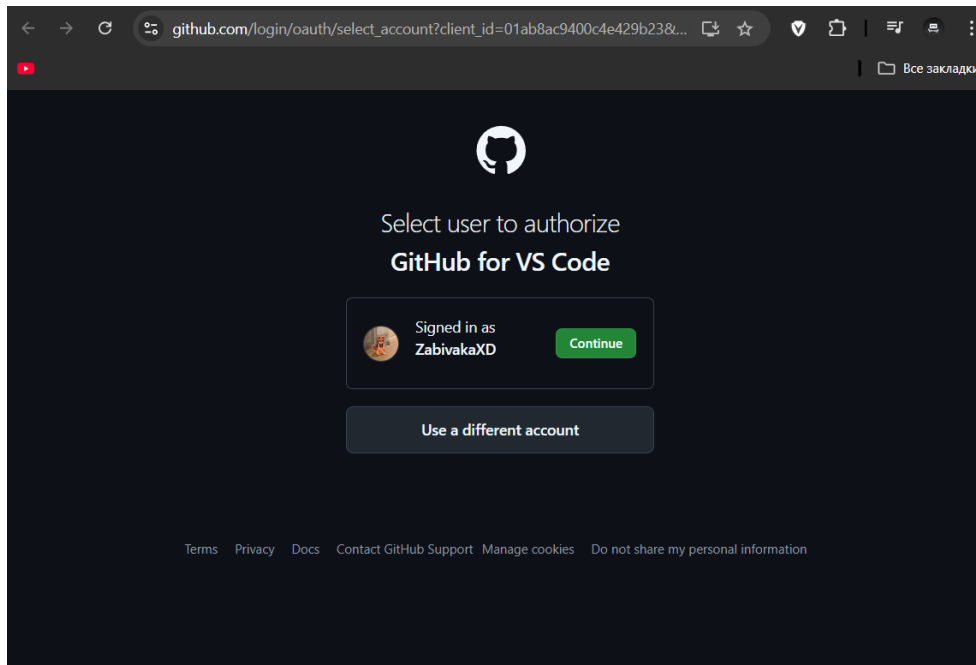
Также можно изменять файл и все изменения будут подсвечиваться их даже можно сравнивать с исходным файлом.



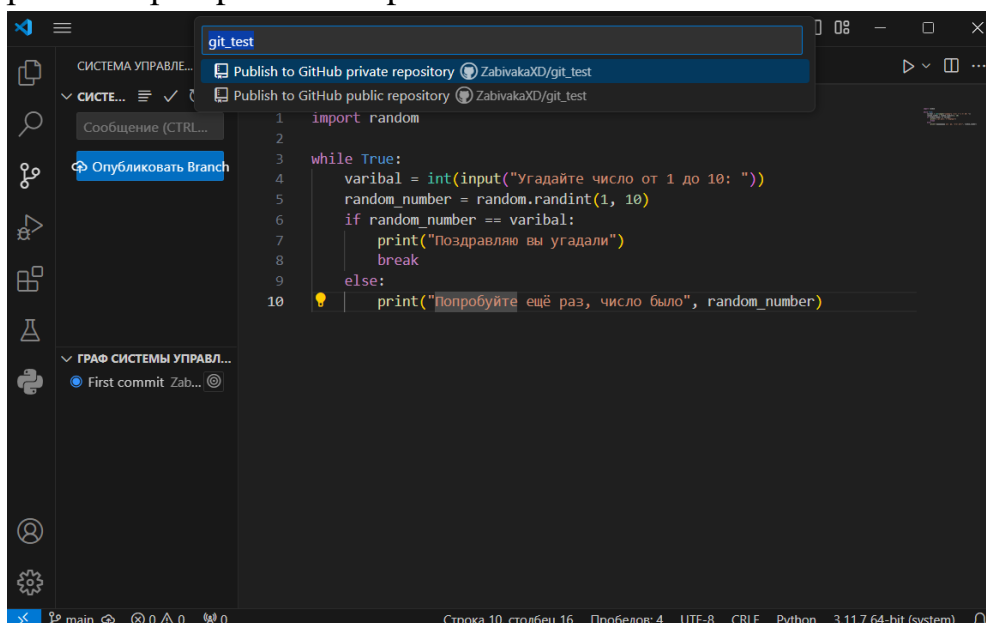
Чтобы репозиторий появился на GitHub его нужно опубликовать.



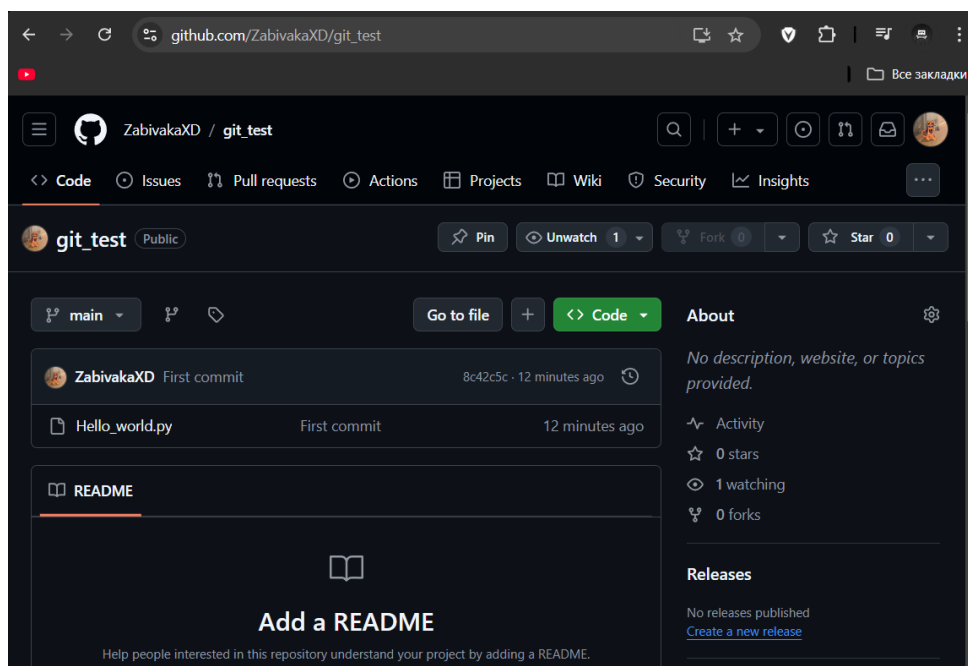
После нажатия нужно подтвердить вход на GitHub, войти в аккаунт.



Подтверждаем подключение VS Code к GitHub и выбираем тип репозитория public или private.



Теперь можно увидеть наш репозиторий в GitHub



Источники:

1. Как пользоваться Visual Studio Code // Timeweb URL:
<https://timeweb.com/ru/community/articles/chto-takoe-visual-studio-code>
(дата обращения: 11.09.2024).
2. Редактор кода Visual Studio Code // Хабр URL:
<https://habr.com/ru/articles/490754/> (дата обращения: 11.09.2024).
3. Сочетания клавиш в Visual Studio // Microsoft URL:
<https://learn.microsoft.com/ru-ru/visualstudio/ide/default-keyboard-shortcuts-in-visual-studio?view=vs-2022> (дата обращения: 11.09.2024).
4. Debugging // Visual Studio Code URL:
<https://code.visualstudio.com/docs/editor/debugging> (дата обращения: 11.09.2024).
5. Улучшаем процесс ведения проекта в Git // Хабр URL:
<https://habr.com/ru/companies/beelinekz/articles/664190/> (дата обращения: 11.09.2024).
6. Using Git source control in VS Code // Visual Studio Code URL:
<https://code.visualstudio.com/docs/sourcecontrol/overview> (дата обращения: 12.09.2024).