

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Сибирский государственный университет геосистем и технологий»
(СГУГиТ)

В.Е. Терещенко

ПРОГРАММИРОВАНИЕ ДЛЯ РЕШЕНИЯ ГЕОДЕЗИЧЕСКИХ И МАРКШЕЙДЕРСКИХ ЗАДАЧ

Лабораторная работ №1
Установка Python и работа в интерактивной рабочей среде. Типы данных.
Математические операции. Операторы

Новосибирск
СГУГиТ
2024

Установка Python

Python – это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью [25, 26], ориентированный на повышение производительности разработчика, читаемости кода (скрипта – от англ. script – сценарий) и его качества, а также на обеспечение переносимости написанных на нём программ [27]. Python распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные [39]. Огромное сообщество энтузиастов уже разработали множество Open Source-модулей для Python, что упрощает процесс разработки всего, что только можно себе представить, и этот процесс не стоит на месте. Таким образом Python стал одним из самых востребованных языков программирования в мире.

Стандартная библиотека включает большой набор полезных переносимых функций, начиная с возможностей для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание web-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на C или C++, при этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках [25]. Существует и специализированный репозиторий программного обеспечения (о котором пойдет речь в данном учебном пособии), написанного на Python – PyPI [45]. Данный репозиторий предоставляет средства для простой установки пакетов в операционную систему и стал де-факто стандартом для Python [46]. По состоянию на 2019 год в нём уже содержалось более 175 тысяч пакетов [45].

Python стал одним из самых популярных языков, он используется в анализе данных, машинном обучении, web-разработке, а также в других сферах, включая разработку игр. За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм.

По состоянию на сентябрь 2022 г. Python занимает первое место в рейтинге TIOBE популярности языков программирования с показателем 15,74 % [48]. «Языком года» по версии TIOBE Python объявлялся в 2007, 2010, 2018, 2020 и 2021 годах [49]. С историей развития, концепцией, философией и методологией Python можно ознакомиться в источниках [xx, xx, xx].

Чтобы начать писать код на Python необходимо установить сам Python. Для этого зайдите на официальный сайт <https://www.python.org/> – рис. 1.

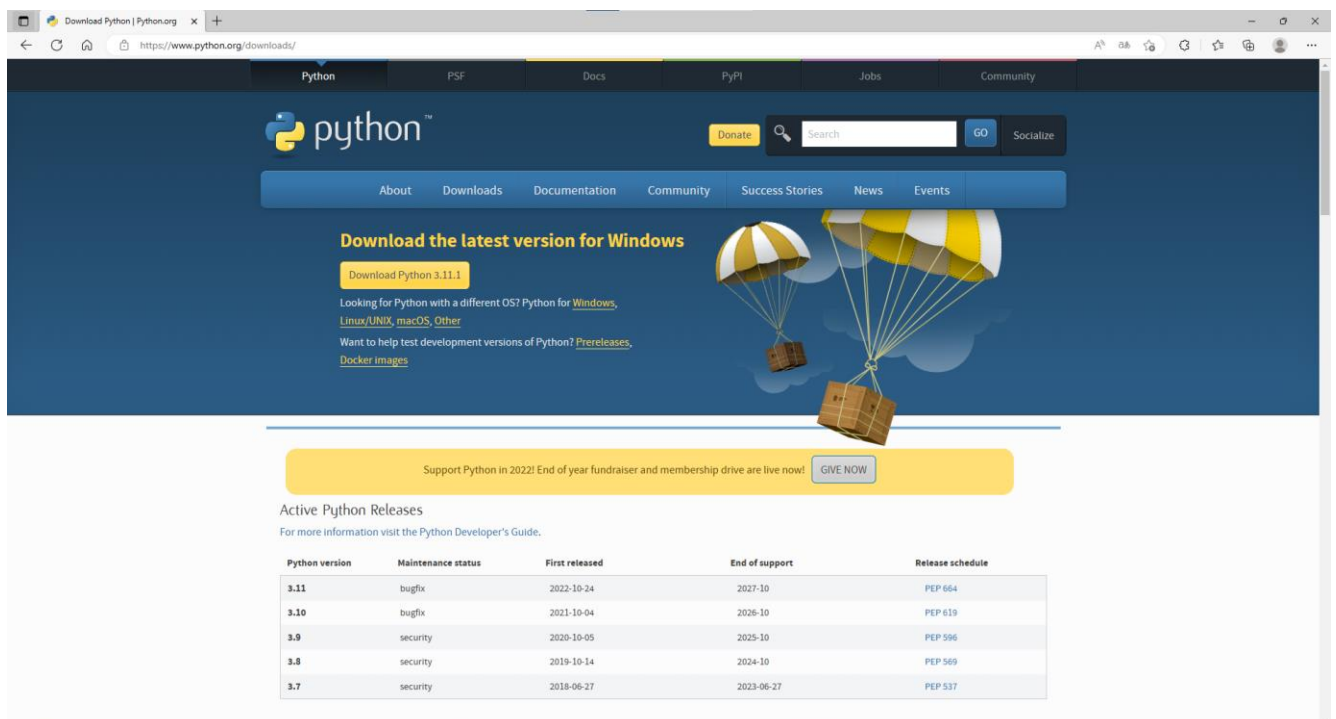


Рис. 1. Официальный сайт Python

Затем в разделе Download выберите актуальную версию программы, кликните на нее и скачайте. В текущей работе приведены скриншоты при работе с Python версии 3.10.8. Запустите загруженный файл и установите Python как обычную программу. На рис. 2 приведен скриншот при установке Python. Обратите внимание на пункт Add Python to PATH. Здесь необходимо поставить галку, чтобы не возникали ошибки в командной строке, и программа по умолчанию взаимодействовала с Windows на системном уровне.

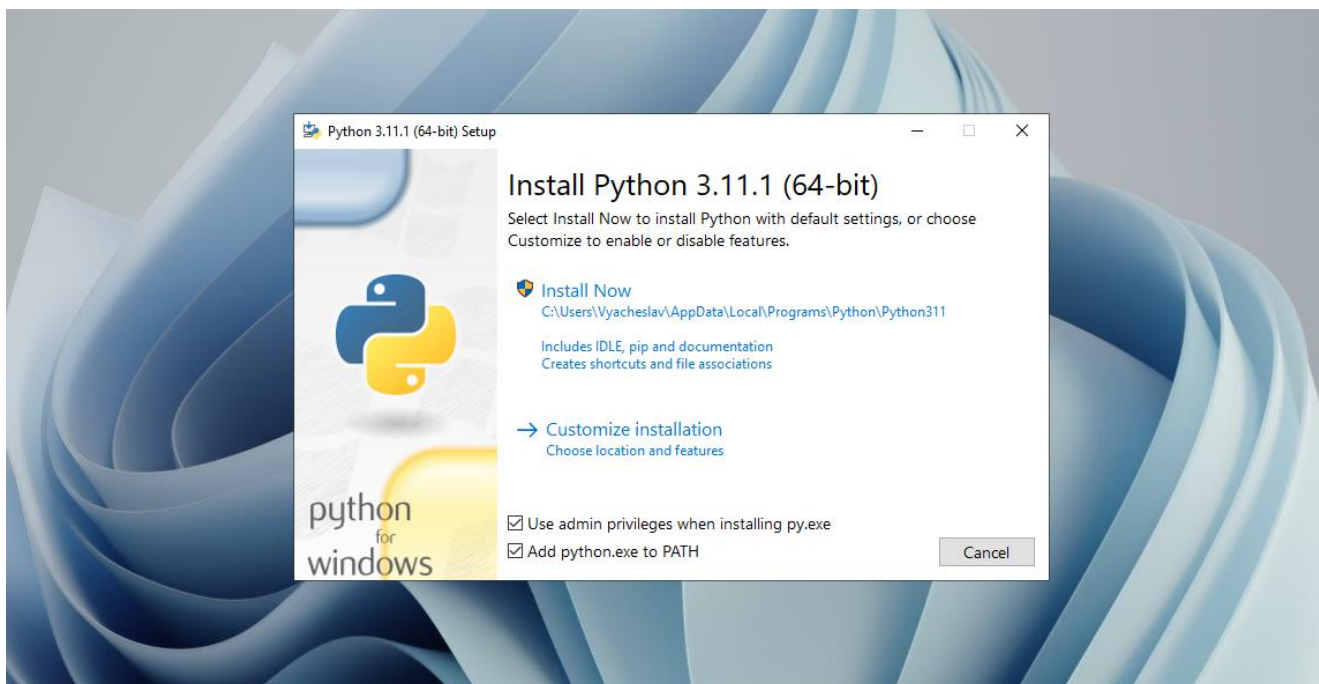


Рис. 2. Окно инсталляции Python

Точно таким образом Python устанавливается на все основные типы операционных систем (Windows, MacOS, Unix-подобные и др.). Поскольку большинство обучающихся работают на компьютерах под управлением операционной системы Windows в данном учебном пособии все примеры, скриншоты и методические указания даны для Windows 10.

Задание 1

Чтобы удостовериться что Python установлен на компьютере выполните следующие действия.

1. Зайдите в командную строку с помощью сочетания клавиш **Win + R**, затем введите «**cmd**» и нажмите **ОК**; (альтернативный способ – это ввести в меню **Пуск** в строке поиск словосочетание «**Командная строка**», или команду «**cmd**»).

2. В открывшемся окне консоли введите «**python**». Если Python установлен, на экране будут выведены строчки с текущей версией программы, датой установки и прочей информацией, как показано на рис. 3. Иначе будет выведена ошибка.

Ошибка может возникнуть если Python установлен без галочки Add Python to Path. Чтобы это исправить, не переустанавливая Python, выполните инструкцию, которая приведена в видео по ссылке ниже. Тайм-код инструкции в видео 7:10. https://www.youtube.com/watch?v=ahe3WpBMdm4&list=PLvoBekrlHDgROfUUHMbrrdsy_b2y2V_rj.

Если Python в командной строке ошибок не возникло, пропустите эту часть.

3. Сделаете скриншот окна с результатом выполнения Вашего кода.

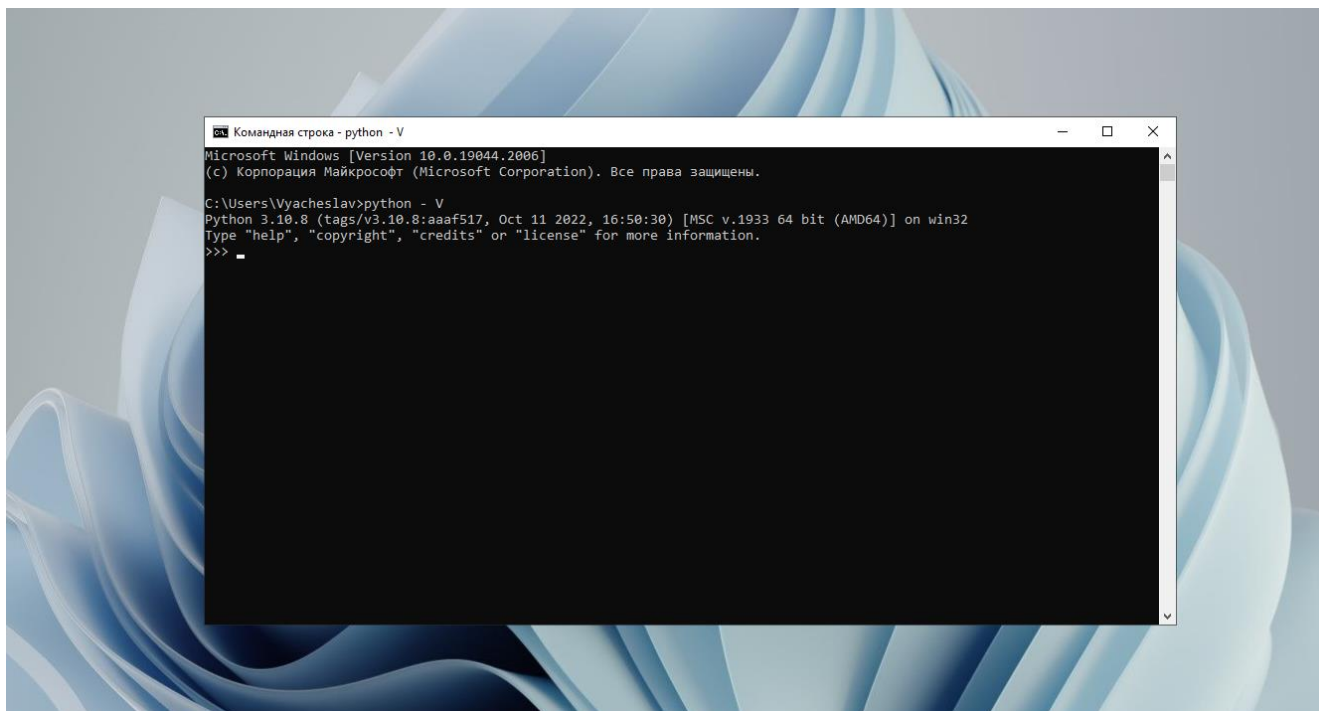


Рис. 3. Окно проверки наличия Python на компьютере

Обратите внимание, что скриншот необходимо делать полноэкранным, всего экрана (без обрезки), чтобы была видна строка пуск с датой и временем выполнения работы. Это важно для проверки отчета о выполнении лабораторной работы, чтобы исключить передачу скриншотов студентами друг другу. Обрезанные скриншоты или скриншоты без наличия строки Пуск с датой и временем могут быть расценены как не верные, и не приняты преподавателем лабораторной работы.

Чтобы сделать скриншот необходимо на клавиатуре нажать на кнопку PrtSc. Скриншот будет скопирован в буфер обмена. Затем Вы можете вставить его в любое необходимое место. Другой способ создать скриншот – воспользоваться сочетанием клавиш **Win + Shift + S** (работает только на Windows 10 и новее). После этого нужно мышкой выберите зону скриншота. Не забудьте включить строку пуск в фрагмент скриншота. Таким способом необходимо создавать скриншоты на протяжении выполнения всех заданий в этой работе и других работах по данному курсу. Не используйте сочетание клавиш Alt + PrtSc, поскольку эта команда создает скриншот только активного окна, не включая в него строку пуск с текущей датой и временем.

Интерактивная среда разработки Python

После установки Python новичкам не ясно, где писать и как запускать код на Python. Существует несколько вариантов.

Первый вариант – работа в стандартной интерактивной оболочке. Она устанавливается одновременно с Python и называется IDLE – Integrated Development and Learning Environment. В переводе – это интегрированная среда разработки и обучения. В некоторых источниках слово «Learning» опускают и интерактивные оболочки именуют аббревиатурой IDE. Никакой существенной разницы здесь нет.

Чтобы открыть стандартную оболочку IDLE Python нажмите **Пуск** и выберите в разделе программы **IDLE**. Откроется окно, в котором уже можно начинать писать команды или тестировать модули. Однако в этом случае можно прописать пока всего одну строку кода. На рис. 4 приведено окно IDLE.

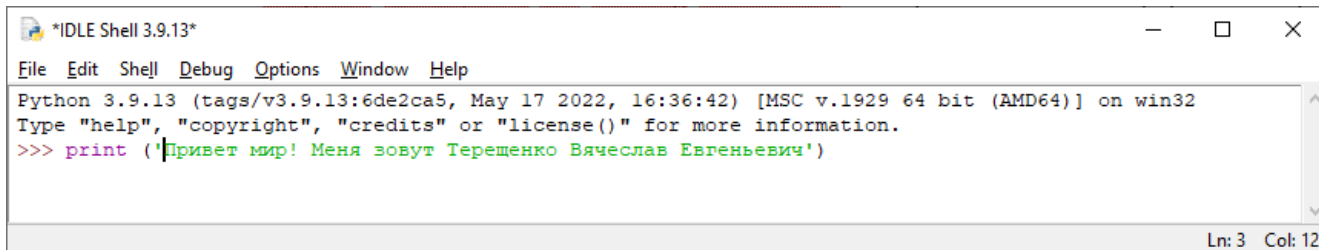


Рис. 4. Окно стандартной интерактивной оболочки IDLE для Python

Данная стандартная IDLE может использоваться как полноценный редактор кода. Чтобы открыть редактор кода нажмите на ленте интерфейса **File**, выберите пункт **New File**. В открытом окне можно вбивать любой код, любой длины. После того как вы напишите код, его нужно запустить, чтобы проверить работоспособность. Для этого

выберете на ленте **Run**, затем **Run model**. IDLE предложит сохранить код, это необходимо сделать, иначе не сохраненный файл не будет запущен. Выберите место для сохранения и название файла и нажмите **ОК**. Файл будет сохранен. Но существует нюанс, после того как вы закроете IDLE при открытии Вашего сохраненного файла ничего не будет отображено. Это происходит потому, что код просто выполнится внутри операционной системе. О том, что нужно открыть окно с результатом исполнения кода этой программе сообщено не было. На данном этапе это можно исправить, введя последней строчкой **input ()**.

Задание 2

Первым заданием для Вас будет написать свою первую программу.

1. Откройте IDLE.
2. На ленте интерфейса выберите **File**, затем **New File**.
3. Введите следующий код: **print('Привет Мир! Меня зовут (укажите здесь свое ФИО)').**
4. В последней строке введите: **input()**.
5. На ленте выберите **Run**, затем **Run model** (можно использовать горячую клавишу F5).
6. Сохраните файл, указав в имени файла свою фамилию, и путь к папке, где он будет находиться. IDLE автоматически запустит его.
7. Сделайте скриншот результата выполнения программы и добавьте его в свой отчет о выполнении работы.

Как принято в этом месте говорить – поздравляю, Вы написали свою первую программу.

Обычно стандартную оболочку IDLE используют новички чтобы тренироваться: писать небольшие скрипты, решать мини задачи и в целом отрабатывать навыки кодинга. Для опытных пользователей существуют значительно более мощные инструменты для создания, отладки и тестирования программ.

Второй вариант – использование внешних редакторов (интегрированных сред разработки – IDE). Их существует много для Python. Например, PyCharm, Spyder, Thonny, Visual Studio, Eclipse, Sublime Text, Visual Studio Code, Atom и много других.

Некоторые разработчики разделяют понятия «Реактор кода» и «Интегрированная среда разработки – IDE». Часто приводят следующие доводы. Редактор кода – это простая легковесная программа, которая похожа на текстовый редактор, только больше адаптированная под написание кода. Она сохраняет проекты в нужном расширении, подсвечивает синтаксис и автоматически проверяет отступы в коде. Интегрированная среда разработки – это более масштабный инструмент, программный комплекс. Помимо редактора в нем есть собственная консоль, инструменты для запуска, тестирования и отладки кода. Через IDE можно создавать крупные проекты, подключать систему контроля версий. Некоторые IDE позволяют устанавливать внутри себя новые фреймворки и библиотеки, не заходя в терминал системы. Также отличия заключаются в таких видах как: подсветка синтаксиса; форматирование и установка отступов; возможность запуска кода; создание, редактирование и сохранение файлов; отладка; кастомизация.

Каждая IDE обладает своими достоинствами и недостатками, об этом написано много мнений, с которыми можно ознакомиться по ссылкам в интернете. В данном учебном пособии методические материалы, скриншоты и задания будут представлены по IDE PyCharm. Необходимо сразу отметить, что для выполнения заданий по данному курсу не обязательно использовать именно PyCharm. Работать можно в любой знакомой среде или в той, которая более удобна. Результат выполнения кода никак не зависит от выбора IDE. Любой файл с расширением .py открывается в любой IDE и может быть отредактирован без каких-либо проблем.

PyCharm – это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик. Чтобы установить PyCharm необходимо скачать дистрибутив с официального сайта <https://www.jetbrains.com/> – рис. 5.

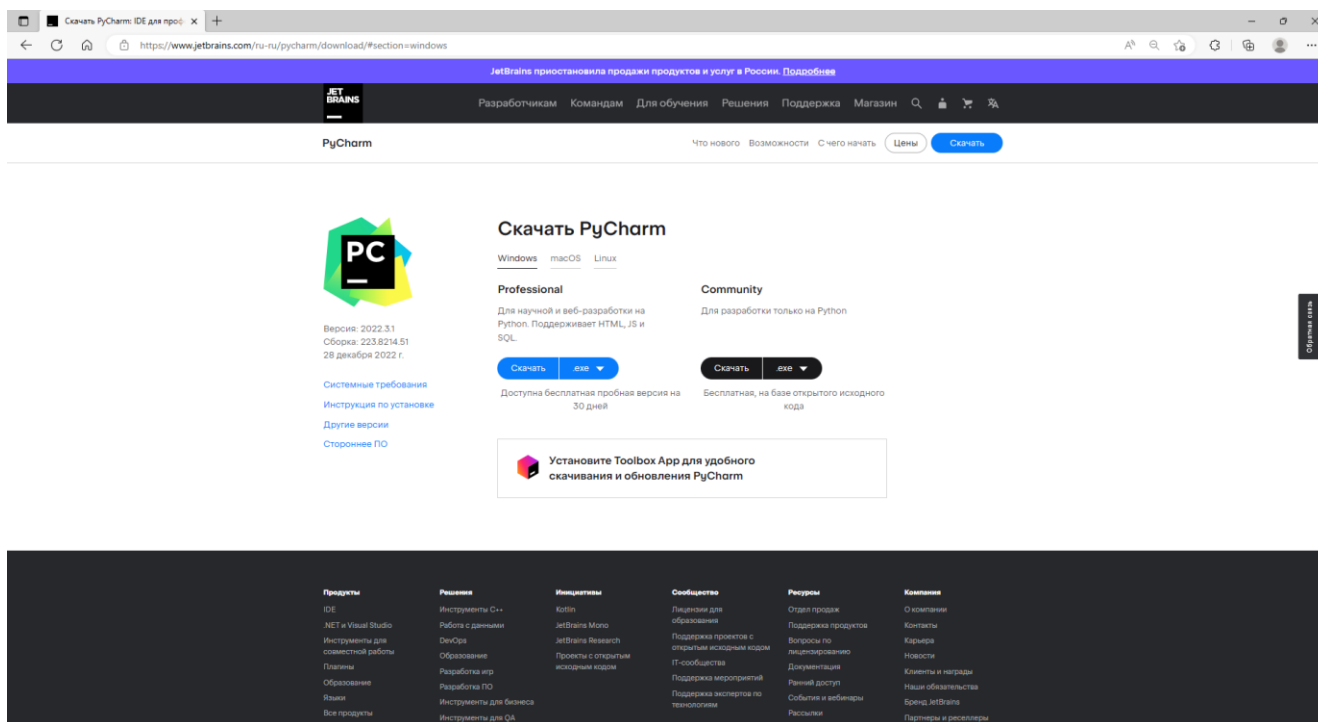


Рис. 5. Официальный сайт JetBrains для скачивания PyCharm

На сайте размещены две версии PyCharm – это Professional и Community. Community – бесплатная лицензионная версия, на базе открытого исходного кода. Предназначена для разработки только на Python. Версия Professional – предназначена для научной и web-разработки на Python и также поддерживает HTML, JS и SQL. Пробный бесплатный период использования составляет 30 дней. После скачивания PyCharm установите его как любую программу для Windows.

Запускается PyCharm также просто, как и любая другая программа на Windows. После запуска чтобы начать писать код необходимо создать новый проект. Кликните на кнопку New project, введите имя проекта и место расположения на диске. Затем нажмите кнопку Create и проект будет создан. В проекте будет создан по умолчанию файл с названием main.py. Он является основным файлом данного проекта и будет отображен в левом окне с другими системными папками. Также его можно найти в папке, которую Вы указали для сохранения проекта.

Русифицированной версии PyCharm на момент начала 2023 года не существует, но в целом в нем, итак, все достаточно понятно для студента, имеющего удовлетворительную оценку по курсу английского языка.

PyCharm это довольно удобный инструмент для программирования, поэтому его часто рекомендуют даже новичкам для изучения программирования. Среди удобных функций можно отметить автозаполнение команд (появляются подсказки для быстрого ввода); автоматическое сохранение кода, то есть не нужно постоянно нажимать сохранить или `ctrl+S` или «дискетку» (сохранение выполняется после каждого ввода символа автоматически); подсветку кода, то есть разными цветами отображаются функции, переменные, списки, текст, комментарии и пр.; простота запуска кода, достаточно нажать на зеленый треугольник справа сверху (run) или комбинацию `Shift + F10` – результат выполнения кода появится в нижнем окне программы.

На данном этапе нужно пояснить как можно запускать код, написанный во внешнем редакторе кода. Существует также два способа. Первый – запуск кода из командной строки. Для этого в командной строке нужно перейти в ту директорию, в которой находится файл кода. Сделайте это следующим образом.

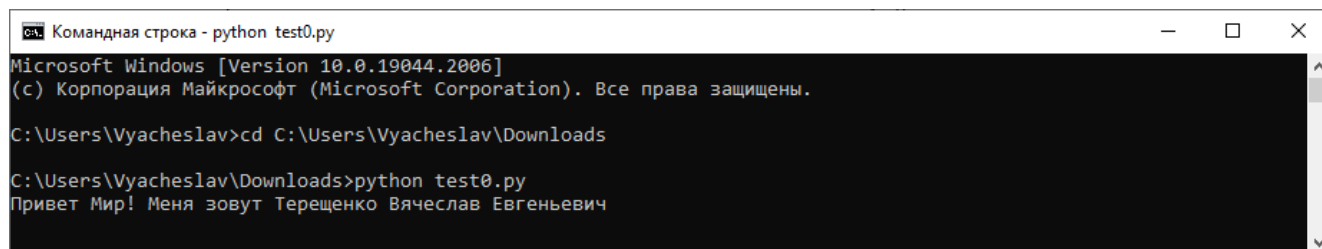
Задание 3

1. Откройте командную строку как делали это вначале с помощью комбинации клавиш **Win + R**, затем введите «`cmd`» и нажмите **OK**.

2. Пропишите путь – директорию, в которой лежит созданный Вами файл кода в строке.

Чтобы в дальнейшем не тратить время на постоянное заполнение директории, перед этим введите команду «`cd`» и символ пробела, как показано в третьей строке сверху на скриншоте на рис. 6. Команда `cd` означает сменить директорию – с англ. change directory – изменить каталог – команда командной строки для изменения рабочего каталога в Unix, DOS и других операционных системах. Если у вас файл кода находится на не системном диске (системный диск, как правило, это диск C), это также нужно указать в командной строке, прописав в отдельной строке букву, обозначающую диск и приписать без пробела символ двоеточие «:», например, «E:». После таких манипуляций Вы будете находиться в нужной директории, из которой можно запускать файл с кодом.

3. Запустите файл кода. Для этого напишите следующую команду: **python** затем наименование своего созданного файла и добавьте расширение **«.py»**. Нажмите Enter. Код будет выполнен. Если в коде есть функция отображения, например, **print**, результат появится в строке ниже.
4. Сделайте скриншот с результатом выполнения кода и вставьте его в отчет о выполнении работы.



```
Командная строка - python test0.py
Microsoft Windows [Version 10.0.19044.2006]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Vyacheslav>cd C:\Users\Vyacheslav\Downloads

C:\Users\Vyacheslav\Downloads>python test0.py
Привет Мир! Меня зовут Терещенко Вячеслав Евгеньевич
```

Рис. 6. Результат выполнения кода в командной строке

Второй способ – запуск кода в самой интегрированной среде IDE, если такая возможность там имеется (в PyCharm это делать можно).

Задание 4

1. Запустите PyCharm. Закройте все окна с рекомендациями, если они появились (но советую реже игнорировать их и знакомиться с подсказками, так как часто они могут облегчить работу программиста).
2. На ленте интерфейса выберите вкладку **File**, затем **Open**. Укажите ту директорию, в которой у Вас находится первый созданный файл и откройте его.
3. В правом верхнем углу PyCharm кликните на зеленый треугольник (или горячие клавиши **Shift + F10**) – запустите файл.

4. Сделайте скриншот всего экрана с результатом выполнения кода программой.

С данного этапа начнем расширять опыт программирования. Введем понятие переменная. Переменная – это своего рода сборник (коробочка) каких-то данных. Эти данные внутри переменной можно изменять, а саму переменную нет, иначе она станет уже другой переменной. Каждая коробочка подписана – имеет свое имя, чтобы понимать, что за данные в ней находятся. Переменные не должны содержать пробелов и никаких специальных символов (например, таких как: `!@#$%^&*()-+=«»№?|/<>.,;:'"[]{}.`). То есть имена переменных могут содержать только буквы, числа и символ подчеркивания (underscore). При этом начинаться переменная на цифру не может. Также все имена переменных в Python регистрочувствительны. То есть переменные «test» и «Test» - являются совершенно разными переменными. Эти правила нужно просто запомнить. Приступим к практике и создадим переменную.

5. Выберите на ленте интерфейса **File**, затем **New project**.

6. В появившемся окне в верхней строке выберите папку локализации проекта.

7. В этом же окне разделе «New environment using» задайте имя конечной папки (введя свою фамилию) после крайнего правого слеша в директории. Остальные настройки оставьте по умолчанию. Раздел «Base interpreter» необходим, чтобы изменить версию Python, если на Вашем компьютере установлены иные версии. Пункт «Previously configured interpreter» необходим, чтобы выбрать параметры, установленные в предыдущем проекте. Чекбокс (галочка) «Create a main.py welcome script» создает в папке файл проекта, по умолчанию названный «main» с расширением «.py». Если чекбокс не будет активирован, придется создать свой файл правой кнопкой мыши, выбрав раздел «New», затем «Python File» и указать его имя.

8. Если PyCharm выдаст окно с вопросом создать проект в данном окне или создать новое, выберите вариант, который Вам удобнее.

9. Постарайтесь осмыслить подсказки-рекомендации PyCharm, которые могут быть выведены при создании проекта.

10. Удалите подсказки и введите переменную с названием **name**. Присвойте ей значение с помощью уже упомянутой функции **input()**. Данная функция позволяет взаимодействовать пользователю с Вашей программой. (с англ. языка переводится как «ввод, введите»). С помощью этой функции можно читать ввод пользователя. Она принимает необязательный строковый аргумент (который выводится в консоли как строка приглашения к вводу) и ожидает, пока пользователь введет ответ и завершит ввод клавишей Enter. Аргументом функции принимаются данные, которые будут введены после слова **input** в круглых скобках. Внутри аргумента функции можно ввести любой текст, например **input('Введите ФИО: ')**. Таким образом программа будет взаимодействовать с пользователем, вывести «Введите ФИО: », и будет ждать установленные значения, чтобы принять их в оперативную память компьютера;

11. Определив переменную, в дальнейшем ее можно вызывать в любой части кода. Сделаем это внутри функции **print**, введя аргументом в круглых скобках (**Привет Мир, меня зовут ' + name**).

12. Сделайте скриншот всего кода и результата его исполнения и поместите его в отчет о выполнении работы.

Важно отметить, что встроенные функции в Python, типа `print`, `input`, `def` и другие, также как и переменные регистрочувствительные и всегда вводятся в нижнем регистре (lower case) – строчными буквами, включая первую. В PyCharm и других IDLE они подсвечиваются цветом.

Задание 5

Задание 5 выглядит следующим образом. Необходимо написать программу, в которой будут две переменные.

1. Первая должна называться **name**. Ей нужно присвоить функцию **input** с возможностью указать ФИО пользователя.
2. Вторая переменная должна называться **age**, в ней должен быть указан возраст равный 25 лет. Без каких-либо функций или возможностей изменить данные **age = 25**.
3. Результат выполнения программы должен быть следующий: «Привет Мир, меня зовут (введенное ФИО), мне 25 лет!». (Вводить необходимо свое собственное ФИО).
4. Сделайте скриншот написанного кода и результата его исполнения. Если возникла ошибка, сделайте скриншот с этой ошибкой.

В данном задании должна возникнуть ошибка. О том, почему так получилось речь пойдет далее.

Типы данных

На данном этапе изучения Python необходимо знать четыре основных типа данных. Это `int`, `float` и `str`. Есть и такие как: `boolean`, `lists`, `tuples`, `dictionaries`, `sets`, но на них пока останавливаться не станем, так как это будет мешать структурированной обработке информации и может вызывать путаницу и неразбериху.

Тип данных `int` – это такой тип, который говорит Python о том, что данное значение – это целое число (от англ. – `integer` – целочисленное значение), то есть не имеет дробной или десятичной части.

Тип данных `float` – это тип данных, который говорит Python, что число является десятичным, имеет после запятой дробную часть. С англ. слово `float` переводится – плавающий, неопределенный.

Тип данных `str` – это тип данных, который представляет значения в тип строка (с англ. `string` переводится – строка). То есть все символы, включая числа будут восприниматься Python как строка, просто символы. В этом случае провести какие-либо математические операции над такими данными не получится. Это все равно, что умножать или вычитать строки между собой. Однако такой инструмент в Python тоже имеется, он называется «Конкатенация». Конкатенация (с лат. языка `concatenatio` – «присоединение, сцепление») – операция склеивания объектов линейной структуры, обычно строк.

В Python не все типы данных совместимы между собой. Допустим есть переменные с тремя типами данных:

```
a = int(3)
b = float(4.5)
c = str(15)
```

Если ввести команду `print(a + b)`, результат вернется 7.5. Но если попробовать сложить переменную `a` и `c` или `b` и `c` – `print(a + c)` или `print(b + c)`, то результат возвращен не будет, программа выдаст ошибку. Тип данных `int` и `float` не совместим с типом данных `str`. На рис. 7 приведен пример сложения переменных с разными типами данных.

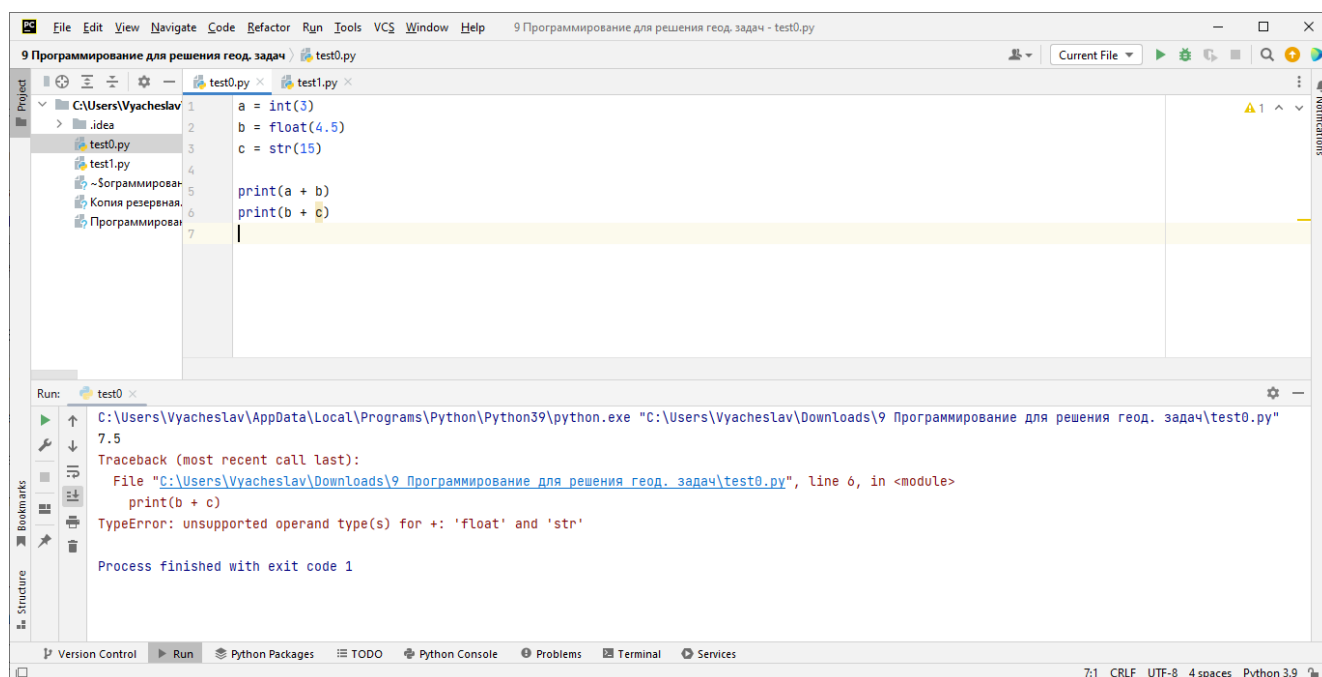


Рис. 7. Фрагмент кода при сложении переменных с разными типами данных

Python устроен таким образом, что если не указывать тип данных, то язык автоматически поймет его тип. Если будет указано целое число, Python усыновит его тип как `int`. Если будет введено число с точкой (запятая в Python, как и в других языках программирования не применяется для обозначения десятичных чисел), то Python определит его тип автоматически как `float`. Если будет введен текст или текст с числами, Python определит тип такой переменной как `str`.

В Python есть еще кое-какие особенности, касающиеся работе со строками. Речь о конкатенации. Допустим есть также три переменные, с типами данных присвоенными Python автоматически:

```
a = 3
b = ("Привет Мир! ")
c = ("Мое имя Вячеслав! ")
```

Если ввести команду `print(b + c)`, результат исполнения программы будет «Привет Мир! Мое имя Вячеслав! ». Но, если ввести команду `print(b * a)` или `print(a * b)`, Python вернет «Привет Мир! Привет Мир! Привет Мир!».

Примечание. В Python не важно введены ли одинарные или двойные кавычки ('', ""). По крайней мере пока, до версии 3.11 это значения не имеет.

Задание 6

Для закрепления материала подумайте и решите, какой результат выполнения следующих команд будет возвращен Python:

1. `print("2" + "3")`
2. `print(int("2" + "3"))`
3. `print(float("123") * int("2"))`
4. `print(float("123" * int("2")))`
5. `print("Hello " * "5")`
6. `print("Hello " * 5)`

Выпишите в отчет результат выполнения всех этих команд в текстовом формате.

Используйте PyCharm или другую IDLE для определения результата данных команд, чтобы проверить себя. Скриншот о выполнении команд представьте в отчете.

Задание 7

1. Исправьте код, написанный задании 5, добавив команду `str()`. Такую операцию в программировании называют тайпкастинг (от англ. type casting – типизация).
2. Используйте только свое собственное ФИО.
3. Сделайте скриншот написанного кода и результата его исполнения. На этот раз никаких ошибок возникнуть не должно.

Математические операции

Любые математические операции в Python выполняются по правилам обычной арифметики. Всегда первое математическое действие Python выполняет в скобках, следующий приоритет выполнения действий – это операция возведения в степень, затем умножение или деление, потом сложение или вычитание, в приоритете выполнения математических действий – слева направо. Математические действия в Python обозначаются символами, приведенными в табл. 1.

Таблица 1. Основные арифметические операторы Python

Оператор	Значение	Пример	Результат
+	Сложение	4 + 6	10
-	Вычитание	9 - 3	6
*	Умножение	5 * 3	15
/	Деление	18 / 2	9
%	Остаток от деления	18 % 4	2
//	Целочисленное деление	18 // 4	4
**	Возведение в степень	4 ** 3	64

Квадратный корень или корень определенной степени из числа в Python не вычисляется с помощью какого-либо символического оператора. Для этого необходимо подгружать дополнительную математическую библиотеку и использовать специальный математический оператор. Об этом будет дана информация ниже.

Задание 8

Для закрепления материала решите следующие простые арифметические задачи. Постарайтесь сделать это без использования PyCharm или другой IDLE. Ответы дайте в печатном виде в отчете о выполнении работы. При необходимости проверьте себя.

1. $(2 + X) ** 2 = 25$
2. $(31 \% X) ** 3 = 27$
3. $X * 2 \% 3 = 1$
4. $82 // 9 ** X = 1$
5. $X + 51 // 10 = 25 // 2$

Операторы присваивания

В Python, как и в любом другом языке программирования, разработчики старались максимально облегчить синтаксис написания кода. Это позволяет не перегружать пользовательский интерфейс и уменьшить рутинные действия IT специалиста. Приведем пример. Допустим существует переменная `test` со значением 5. А в ходе выполнения программы эту переменную нужно превратить в число 10. Как это можно сделать?

```
test = 5
test = test + 5
print(test)
```

результат будет равен 10.

Следует отметить, что Python выполняет код, как другие языки, слева направо и сверху вниз. То есть, если ниже по коду переменная изменяет свое значение, Python оставит в оперативной памяти компьютера нижнее значение и будет выполнять команды далее постепенно – строчка за строчкой.

Вернемся к операторам присваивания. Дана та же переменная:

```
test = 5
test += 5
print(test)
```

результат также будет равен 10.

Как арифметические операторы, операторы присваивания можно структурировать. В табл. 2 приведены основные операторы присваивания Python.

Таблица 2. Основные операторы присваивания Python

Оператор	Пример	Тот же результат
=	<code>x = 5</code>	<code>x = 5</code>
+=	<code>x += 6</code>	<code>x = x + 6</code>
-=	<code>x -= 3</code>	<code>x = x - 3</code>
*=	<code>x *= 3</code>	<code>x = x * 3</code>
/=	<code>x /= 2</code>	<code>x = x / 2</code>
%=	<code>x %= 4</code>	<code>x = x % 4</code>
//=	<code>x //= 4</code>	<code>x = x // 4</code>
**=	<code>x **= 3</code>	<code>x = x ** 3</code>

Задание 9

Для закрепления материала решите следующие простые задачи. Постарайтесь сделать это без использования PyCharm или другой IDLE. Ответы дайте в печатном виде в отчете о выполнении работы. При необходимости проверьте себя.

- | | | | | |
|--|--|---|---|--|
| 1.
<code>a = 5</code>
<code>b = 2</code>
<code>a += b</code>
<code>print(a)</code> | 2.
<code>c = 2</code>
<code>d = 11</code>
<code>d //= c</code>
<code>print(d)</code> | 3.
<code>f = 17</code>
<code>f %= 4</code>
<code>print(f)</code> | 4.
<code>g = "test"</code>
<code>g *= 5</code>
<code>print(g)</code> | 5.
<code>h = "test2"</code>
<code>h /= 5</code>
<code>print(h)</code> |
|--|--|---|---|--|

Комментирование строк

Комментирование в Python, как и в любом другом языке программирования необходимо для того, чтобы описать для себя или для того, кто будет читать код, то, что делает тот или иной участок кода, или строка.

Комментирование обозначается символом решетки # (хэш, шарп, диз). Все символы или данные, которые идут правее от решетки Python игнорирует. При этом символы слева в расчет берутся. Это касается только той

строки, в которой символ установлен. Со следующей строки Python работает в нормальном режиме. Обычно IDLE подсвечивают серым или полупрозрачным цветом тот участок кода, который закомментирован.

В PyCharm существует сочетание горячих клавиш, чтобы закомментировать всю строку целиком, или целиком несколько строк. Используйте сочетание клавиш **Ctrl + /**, при установленном курсоре на строке, которую необходимо закомментировать, или выделенных строках. Обратите внимание, что данное сочетание клавиш в PyCharm работает только на английской раскладке клавиатуры.

Также в Python существует другой способ задания многострочных комментариев. Можно ввести тройные кавычки, написать текст, который нужно закомментировать, и закрыть тройные кавычки. То, что внутри них обработано не будет. При том здесь тоже не важно они будут кавычки одинарные или двойные, главное, чтобы если открыты три двойные кавычки, то и закрыты должны быть именно три двойные. Если открыты три одинарные, то и закрыты должны быть именно три одинарные кавычки.

В Python существует еще одна опция связанная с комментированием кода, она называется экранирование. Обозначается символом обратной косой черты \ (бэкслеш). Например, необходимо вывести фразу: «Пароль для доступа к интернет соединению "12345678" никому его не сообщайте!». Если написать код таким образом:

```
print("Пароль для доступа к интернет-соединению "12345678", никому его не сообщайте!")
```

то PyCharm выдаст ошибку. Пример представлен на рис. 8.

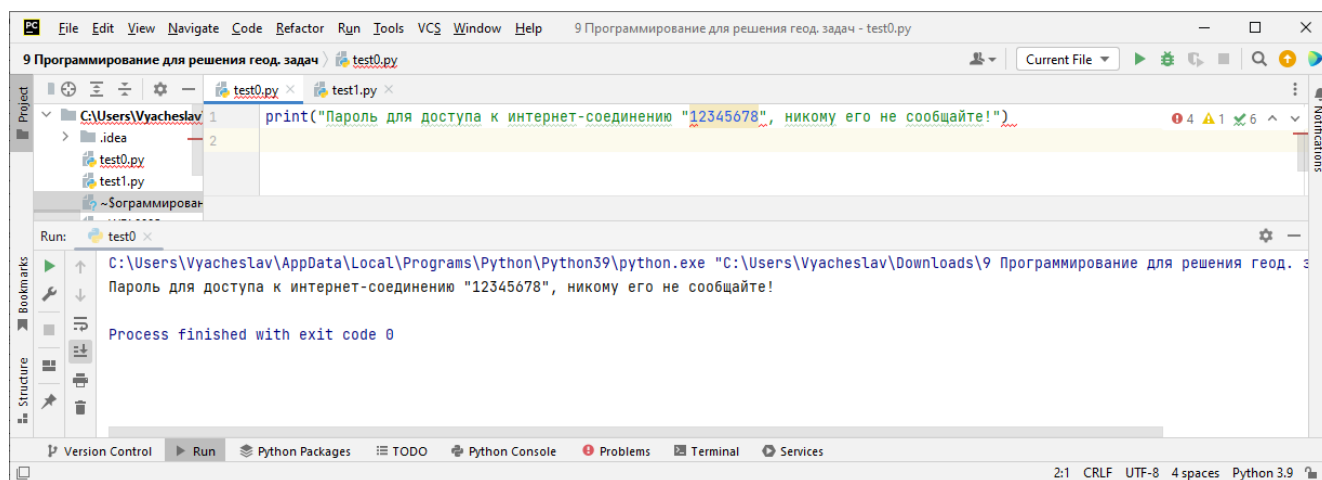


Рис. 8. Пример ошибки в коде

Такая ошибка возникает из-за того, Python воспринимает вторую кавычку в коде, как то, что закрывает первую открытую кавычку. Следовательно символы, идущие после второй кавычки, уже становятся неизвестными для Python. Чтобы такой ошибки не возникало применяют экранирование – ставят символ «\» (бэкслеш) перед кавычкой, что позволяет Python этот символ игнорировать. Пример исправленной версии кода на рис. 9.

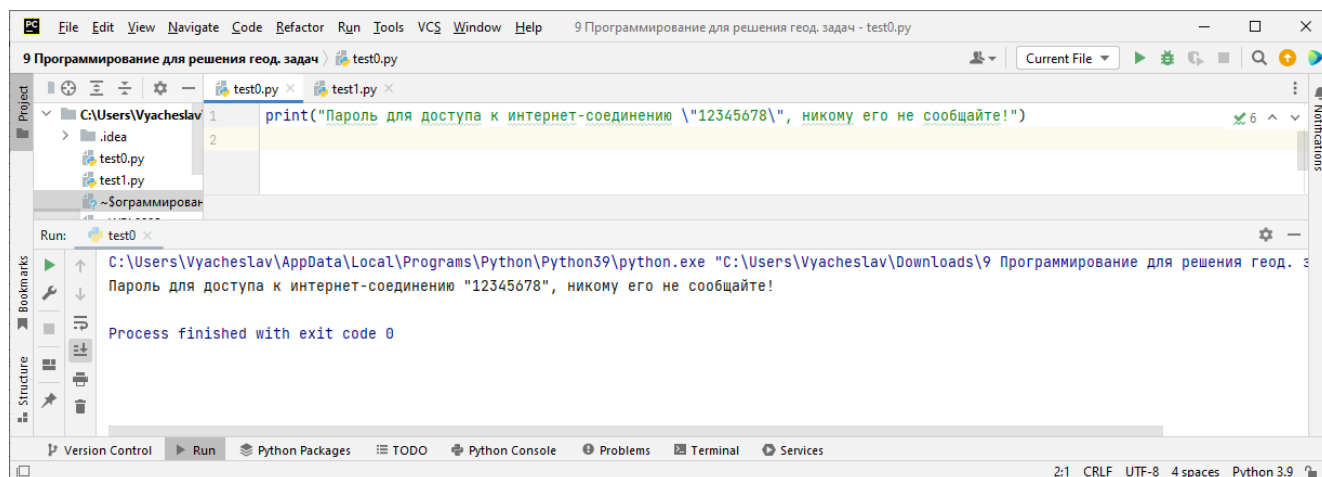


Рис. 9. Пример исправленной ошибки в коде

Также данную проблему можно решить с помощью использования одинарных и двойных кавычек. Внутри одного типа кавычек заключить другой тип. Например, внутри двойных кавычек заключить одинарные, или наоборот. Python это поймет, и никакой ошибки в коде не возникнет.

Операторы сравнения

На текущий момент мы располагаем тремя типами переменных, это int, float и str. Сейчас познакомимся с еще одним типом, это тип Boolean. Этот тип переменной имеет всего два значения – это константа True и константа False. Регистр имеет значение. В переводе с англ. – True – правда, False – ложь. Других смысловых значение в этих переменных больше нет.

Чтобы узнать или указать является ли одна переменная больше, меньше равной или не равной другой переменной в Python, как и в других языках программирования, предусмотрены операторы сравнения. В табл. 3 приведены все операторы сравнения, которые применяются в Python.

Таблица 3. Операторы сравнения в Python

Оператор	Значение	Пример
==	Равно	x == y
!=	Не равно	x != y
>	Больше	x > y
<	Меньше	x < y
>=	Больше или равно	x >= y
<=	Меньше или равно	x <= y

Использование одного символа = (равно) означает, что Python присваивает данные переменной. Использование символов == (двойное равно) означает, что Python сравнивает данные (как будто спрашивает, а равны ли эти переменные, данные?).

Например, существует переменная X и переменная Y. Допустим X = 10, Y = 5. Чтобы узнать, больше ли X, чем Y, введем команду:

```
print(X == Y)
```

В результате выполнения кода Python выдаст результат «False». Данные, которые будут выведены, относятся к типу Boolean. Если X и Y будут иметь одинаковые значения, например X = 5, Y = 5, то результат будет выведен – «True». Таким же образом Python понимает все те операторы сравнения, которые указаны в табл. 3.

Задание 10

Каким будет результат следующих выражений? Постарайтесь сделать это без использования PyCharm или другой IDLE. Ответы дайте в печатном виде в отчете о выполнении работы. При необходимости проверьте себя.

1. X = 20 Y = 19 print(X != Y)	2. X = 10 Y = 10.0 print(X > Y)	3. X = 10 Y = 10.0 Z = 3 print(X == Y > Z)	4. X = 10 Y = 10.0 Z = 3 print(X = Z)	5. X = "testA" Y = "testB" print(X < Y)	6. X = 5.10 Y = 5.1 print(X == Y)
---	--	--	---	--	--