

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.1
дисциплины «Искусственный интеллект в профессиональной сфере»

Выполнил:
Мирошниченко Кирилл
Владимирович,
3 курс, группа ЭНЭ-б-о-22-1,
11.03.04 «Электроника и
наноэлектроника», направленность
(профиль) «Промышленная
электроника», очная форма обучения

(подпись)

Проверил:
Воронкин Роман Александрович,
доцент

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема работы: основы языка Python.

Цель работы: исследование процесса установки и базовых возможностей языка Python версии 3.x.

Аппаратура и материалы: ПК, операционная система Windows 10, Git, браузер для доступа к web-сервису GitHub, Anaconda, PyCharm.

Ход работы:

1. Установил Python.

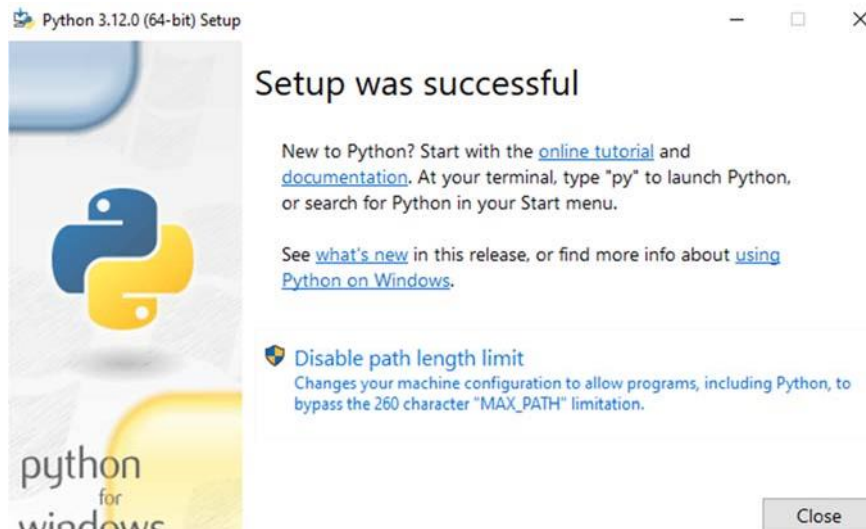


Рисунок 1 – Python

2. Установил Anaconda.

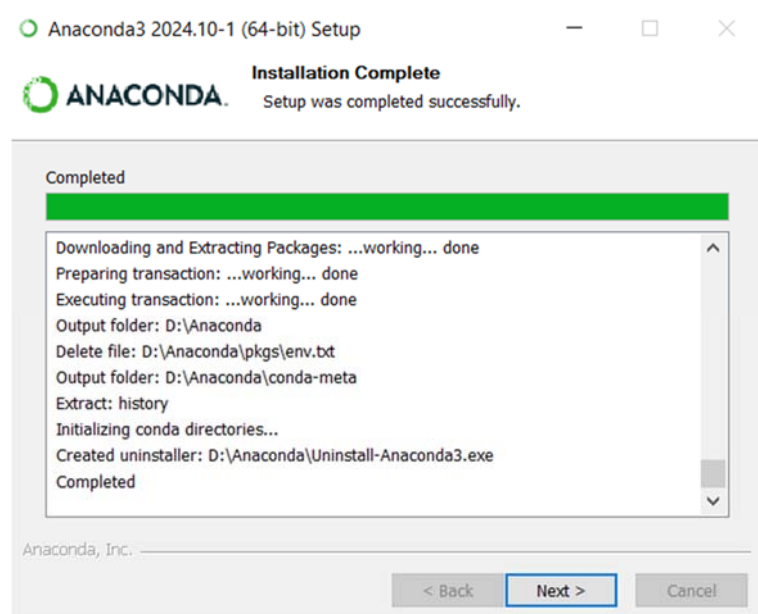


Рисунок 2 – Anaconda

3. Установил PyCharm Community Edition.

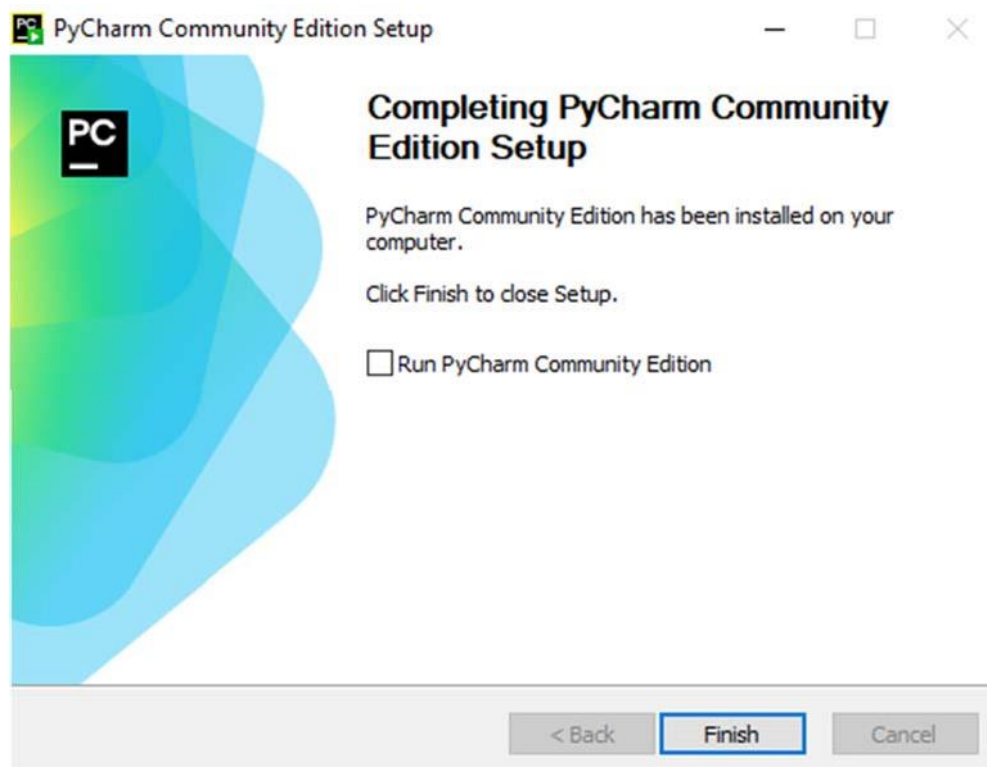


Рисунок 3 – PyCharm

4. Изучил теоретический материал работы. На основе полученных знаний создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и выбранный мной язык программирования (python).

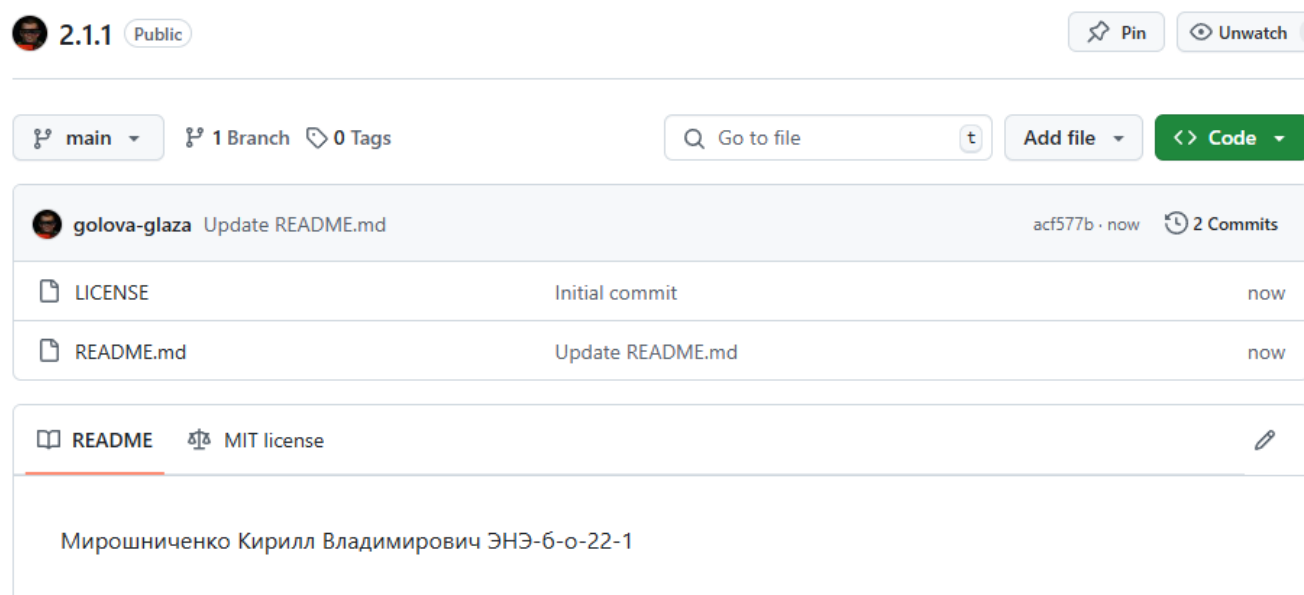


Рисунок 4 – Новый репозиторий

6. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm. Создал проект PyCharm в папке репозитория.

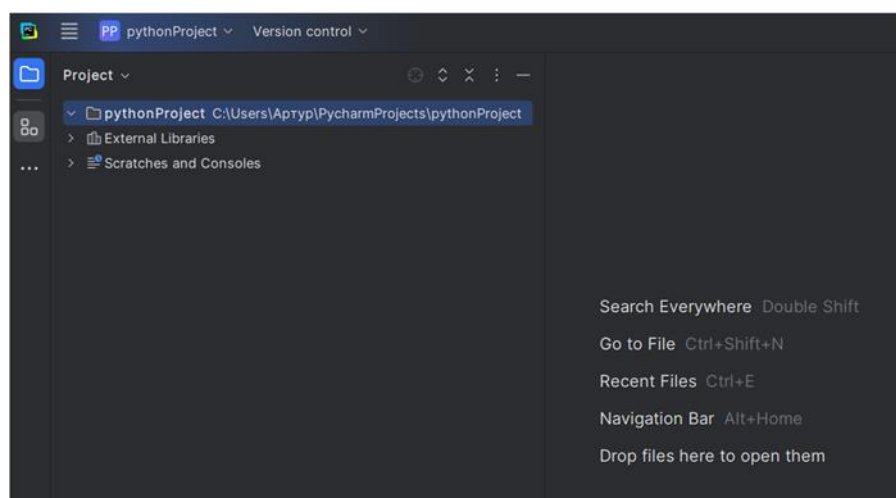


Рисунок 6 – Проект

8. Напишите программу (файл *user.py*), которая запрашивала бы у пользователя:

- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?")

После этого выводила бы три строки:

Рисунок 7 – Задача 1

```
>>> def main():
...     name = input("Введите ваше имя: ")
...     age = input("Введите ваш возраст: ")
...     location = input("Введите ваше место жительства: ")
...     print("\nИнформация о пользователе:")
...     print(f"Имя: {name}")
...     print(f"Возраст: {age}")
...     print(f"Место жительства: {location}")
...     if __name__ == "__main__":
...         main()
...
Введите ваше имя: >? Кирилл
Введите ваш возраст: >? 20
Введите ваше место жительства: >? Ставрополь

Информация о пользователе:
Имя: Кирилл
Возраст: 20
Место жительства: Ставрополь
```

Рисунок 8 – Решение 1

9. Напишите программу (файл *arithmetic.py*), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

Рисунок 9 – Задача 2

```
Python Console x +
>>> def main():
...     correct_answer = 4 * 100 - 54
...     user_input = input("Решите пример 4 * 100 - 54: ")
...     user_answer = int(user_input)
...     print("Ваш ответ:", user_answer)
...     print("Правильный ответ:", correct_answer)
>>> if __name__ == "__main__":
...     main()
...
Решите пример 4 * 100 - 54: >? 346
Ваш ответ: 346
Правильный ответ: 346

>>> |
```

Рисунок 10 – Решение 2

9. Решил задачу, представленную на рисунке 11.

10. Запросите у пользователя четыре числа (файл *numbers.py*). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.

Рисунок 11 – Задача 3

```
Python Console
... num1 = float(input("Введите первое число: "))
... num2 = float(input("Введите второе число: "))
... num3 = float(input("Введите третье число: "))
... num4 = float(input("Введите четвертое число: "))
... sum1 = num1 + num2
... sum2 = num3 + num4
... result = sum1 / sum2
... print(f"Результат: {result:.2f}")
... if __name__ == "__main__":
...     main()
...
Введите первое число: >? 8
Введите второе число: >? 0
Введите третье число: >? 3
Введите четвертое число: >? 5
Результат: 1.00
```

Рисунок 12 – Решение 3

10. Решил индивидуальную задачу, вариант 10

10. Выпуклый четырехугольник задан координатами своих вершин. Найти площадь этого четырехугольника как сумму площадей треугольников.

Рисунок 13 – Индивидуальное задание

Ответы на вопросы:

1. Опишите основные этапы установки Python в Windows и Linux.

Основные этапы установки Python в Windows:

Скачивание дистрибутива. Нужно перейти на официальный сайт Python

```
Python Console
... return abs((x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / 2)
... def quadrilateral_area(x1, y1, x2, y2, x3, y3, x4, y4):
...     area1 = triangle_area(x1, y1, x2, y2, x3, y3)
...     area2 = triangle_area(x1, y1, x3, y3, x4, y4)
...     return area1 + area2
... x1, y1 = 0, 0
... x2, y2 = 4, 0
... x3, y3 = 4, 3
... x4, y4 = 0, 3
... area = quadrilateral_area(x1, y1, x2, y2, x3, y3, x4, y4)
... print("Площадь четырехугольника:", area)
Площадь четырехугольника: 12.0
```

и выбрать Windows на вкладке «Скачать».

Выбор версии. В выпадающем меню по умолчанию появится последняя версия для операционной системы пользователя.

Скачивание установочного файла. Для Windows скачивается файл в формате *.exe. 1

Запуск установочного файла. Нужно дважды щёлкнуть по скачанному файлу, чтобы запустить мастер установки Python.

Выбор опций установки. В нижней части окна установщика следует проверить чекбокс «Добавить Python в PATH». Затем выбрать «Установить сейчас» для стандартной установки или «Настроить установку», чтобы выбрать конкретные функции и место установки.

Завершение установки. После завершения установки появится сообщение об успешном завершении.

Проверка установки. Для этого нужно открыть окно «Выполнить» сочетанием клавиш Win + R и ввести cmd. В открывшейся командной строке ввести команду `py` или `python`, чтобы запустить интерпретатор.

Основные этапы установки Python в Linux:

Определение пакетного менеджера дистрибутива. Например, для Debian, Ubuntu и основанных на них дистрибутивов это apt, для CentOS, Red Hat, Fedora — rpm, для ARCH Linux — pacman, для OpenSUSE и SUSE Linux — zypper.

Обновление списка доступных пакетов. Для этого нужно запустить терминал и обновить список доступных пакетов, используя пакетный менеджер apt с командой `sudo apt update`.

Установка Python. Далее нужно использовать команду пакетного менеджера для установки пакета Python3, например, `sudo apt install python3 -y`.

Установка pip. Теперь необходимо установить pip — пакет для загрузки модулей Python, для этого нужно ввести команду `sudo apt install python3-pip -y`.

Проверка правильности установки. Для этого нужно проверить, установлен ли Python правильно, для этого стоит написать в консоль команду `python3`.

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Основное отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта, заключается в том, что Anaconda включает в себя дополнительные инструменты и пакеты

3. Как осуществить проверку работоспособности пакета Anaconda?

Проверить открытие Anaconda Navigator. Если установка прошла успешно, навигатор откроется автоматически. Если нет, можно открыть его вручную.

Проверить установку через интерфейс командной строки (CLI). Для этого нужно открыть CLI для своей операционной системы.

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?
Откройте File -> Settings -> выберите проект -> Python Interpreter

В строке выбора интерпретатора нажмите “Show all”. Нажмите на +. Создайте новое виртуальное окружение, в качестве базового интерпретатора укажите путь к файлу `python.exe`.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Использовать горячие клавиши. Для Windows и Linux это `Ctrl + Shift + F10`, Внизу раскроется вкладка Run, в которой отобразится результат выполнения.

Нажать на зелёную стрелку в панели инструментов. Также можно перейти в меню Run и там нажать на подпункт Run «main».

Нажать на значок возле номера строки. Во всплывающем окне выбрать пункт Run.

6. В чем суть интерактивного и пакетного режимов работы Python?

Интерактивный режим — это режим запуска кода Python, в котором пользователь построчно вводит команды на исполнение. Интерпретатор

понимает, что если введена однострочная команда, которая не требует продолжения, то её необходимо выполнить сразу.

Пакетный режим — это способ запуска Python-скриптов, при котором скрипт можно запустить обращением к сохранённому файлу с кодом через терминал.

7. Почему язык программирования Python называется языком динамической типизации?

Python является языком с динамической типизацией. А это значит, что переменная не привязана жестко к определенному типу. Тип переменной определяется исходя из значения, которое ей присвоено. Так, при присвоении строки в двойных или одинарных кавычках переменная имеет тип `str`. При присвоении целого числа. Python автоматически определяет тип переменной как `int`.

8. Какие существуют основные типы в языке программирования Python?

Числовые типы. Целые числа (`int`), числа с плавающей точкой (`float`) и комплексные числа (`complex`).

Строки (`str`). Последовательности символов, заключённых в кавычки.

Списки (`list`). Упорядоченные коллекции элементов, которые могут быть разных типов. Они мутабельны, то есть их элементы можно изменять.

Кортежи (`tuple`). Похожи на списки, но они неизменяемы, то есть их элементы не могут быть изменены после создания.

Множества (`set`). Неупорядоченные коллекции уникальных элементов.

Словари (`dict`). Коллекции пар «ключ-значение». Они неупорядочены и мутабельны.

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

Создание объекта в памяти — это выделение и инициализация памяти для хранения значений полей объекта. Адрес объекта — это номер ячейки, начиная с которой объект размещается в памяти.

Устройство объекта включает такие атрибуты, как тип, адрес памяти, присвоенное значение и имя. Объект может включать в себя другие объекты, например, объект класса круга Circle может содержать объект класса точки Point как свой центр.

Процесс объявления новых переменных заключается в том, что перед использованием в программе любой объект данных должен быть объявлен. Оператор объявления сообщает компилятору информацию об идентификаторах и типах данных, назначаемых объектам, и о предполагаемом времени жизни этих объектов.

Работа операции присваивания заключается в том, что значение переменной задаётся (изменяется) с помощью оператора присваивания. Общий вид оператора присваивания: имя_переменной = выражение. Выражение может быть как простой одиночной константой (например, числом), так и сложной комбинацией переменных, операторов и констант.

10. Как получить список ключевых слов в Python?

Чтобы получить список ключевых слов в Python, можно использовать библиотеку keyword. Вот пример кода:

```
import keyword  
s = keyword.kwlist
```

11. Каково назначение функций id() и type()?

id() возвращает идентификатор объекта в виде целого числа. Это число обычно относится к месту хранения объекта в памяти. type() возвращает тип объекта.

12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым типам относятся: целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set). К изменяемым типам относятся: списки (list), множества (set), словари (dict). Изменяемый объект можно изменить после его создания, а неизменяемый — нет

13. Чем отличаются операции деления и целочисленного деления?

Операции деления и целочисленного деления в языке Python отличаются результатом.

При делении с использованием оператора `/` результатом будет точное число с плавающей точкой. Целочисленное деление выполняется с помощью оператора `//` и позволяет получить целую часть операции. При таком округлении дробная часть отбрасывается, а целая часть округляется в меньшую сторону.

14. Какие имеются средства в языке Python для работы с комплексными числами?

Создание комплексного числа. Для этого можно использовать функцию `complex(a, b)`, в которую в качестве первого аргумента передаётся действительная часть, в качестве второго — мнимая. Либо записать число в виде $a + bj$. Операции с комплексными числами. Их можно складывать, вычитать, умножать, делить и возводить в степень

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

Назначение модуля `math` в Python — предоставление доступа к математическим функциям для работы с действительными числами. Он состоит из различных математических констант и функций, которые можно использовать после импорта модуля

16. Каково назначение именованных параметров `sep` и `end` в функции `print()`? Параметр `sep` определяет символ или строку, которая будет использоваться для разделения нескольких значений, переданных в `print()`. По умолчанию `sep` установлен как пробел.

Параметр `end` определяет, что будет напечатано после всех переданных значений. Стандартное значение `end` — символ новой строки (`\n`), что означает перевод строки после выполнения `print()`

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Метод `format()` — это более гибкий и мощный способ форматирования строк в Python. Он позволяет использовать именованные аргументы и указывать порядок, в котором они должны отображаться в строке.

Другие средства форматирования строк в Python:

Оператор `%`. Позволяет вставлять значения в строки, используя знак процента и спецификаторы формата.

f-строки. Представляют собой строковые литералы, которые начинаются с префикса `f` или `F`. Внутри фигурных скобок можно указывать выражения Python.

Шаблонные строки. Используют класс `Template` из стандартной библиотеки. Позволяют создавать шаблоны, в которые можно вставлять значения.

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

Для ввода с консоли значения целочисленной переменной в Python нужно использовать функцию `input()` в сочетании с функцией `int()` для преобразования введенной строки в целое число. Пример кода: `int_number = int(input())`. Для ввода вещественной переменной используется функция `float()`. Пример кода: `float_number = float(input())`. При использовании `input()` важно быть внимательным, так как некорректный ввод может привести к ошибкам. Например, попытка преобразовать в целое число строку, которая не может быть интерпретирована как число, вызовет ошибку `ValueError`. Также можно сократить программный код, если преобразование типов выполнить в тех же строках кода, где вызывается функция `input()`.

Вывод: в ходе выполнения работы был исследован процесс установки и базовых возможностей языка Python версии 3.x