

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**  
**дисциплины «Основы программной инженерии»**

Выполнил:

Матвеев Александр Иванович  
1 курс, группа ПИЖ-б-о-22-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка и  
сопровождение программного  
обеспечения», очная форма обучения

\_\_\_\_\_  
(подпись)

Проверил Воронкин Роман Александрович

\_\_\_\_\_  
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** основы языка Python.

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

Ход работы.

## 1. Создание нового репозитория с лицензией MIT.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

**Owner \*** SashkaHacker / **Repository name \*** laba3  
✅ laba3 is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-happiness](#) ?

**Description** (optional)  
Выполнение лабораторной работы №3 по дисциплине: основы программной инженерии

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**  
.gitignore template: Python  
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**  
License: MIT License  
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

**Create repository**

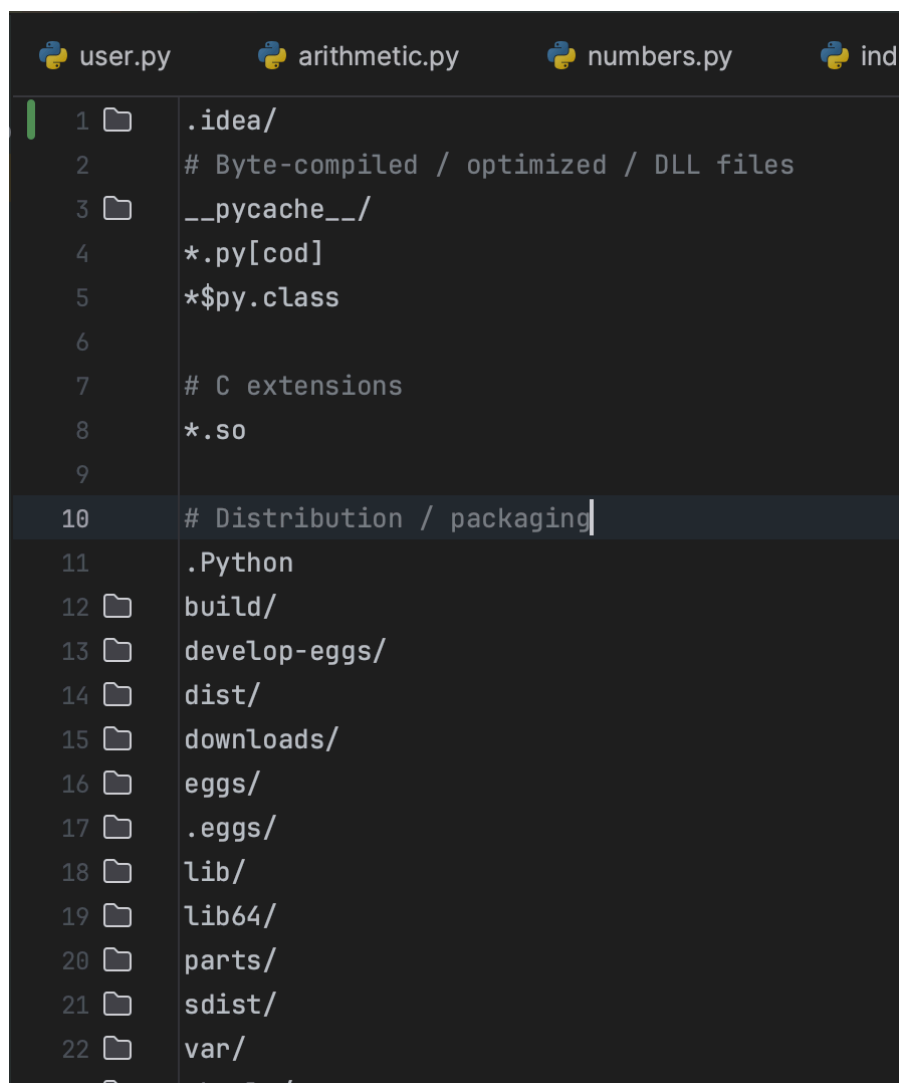
Рисунок 1 – Создание репозитория

## 2. Клонировал репозиторий на рабочий ПК.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/Documents/GitHub
$ git clone https://github.com/SashkaHacker/laba4.git
Cloning into 'laba4'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 28 (delta 10), reused 12 (delta 3), pack-reused 0
Receiving objects: 100% (28/28), 7.29 KiB | 1.82 MiB/s, done.
Resolving deltas: 100% (10/10), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.



```
user.py arithmetic.py numbers.py ind
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

Рисунок 3 – Файл .gitignore

4. Создание PyCharm проекта в репозитории.

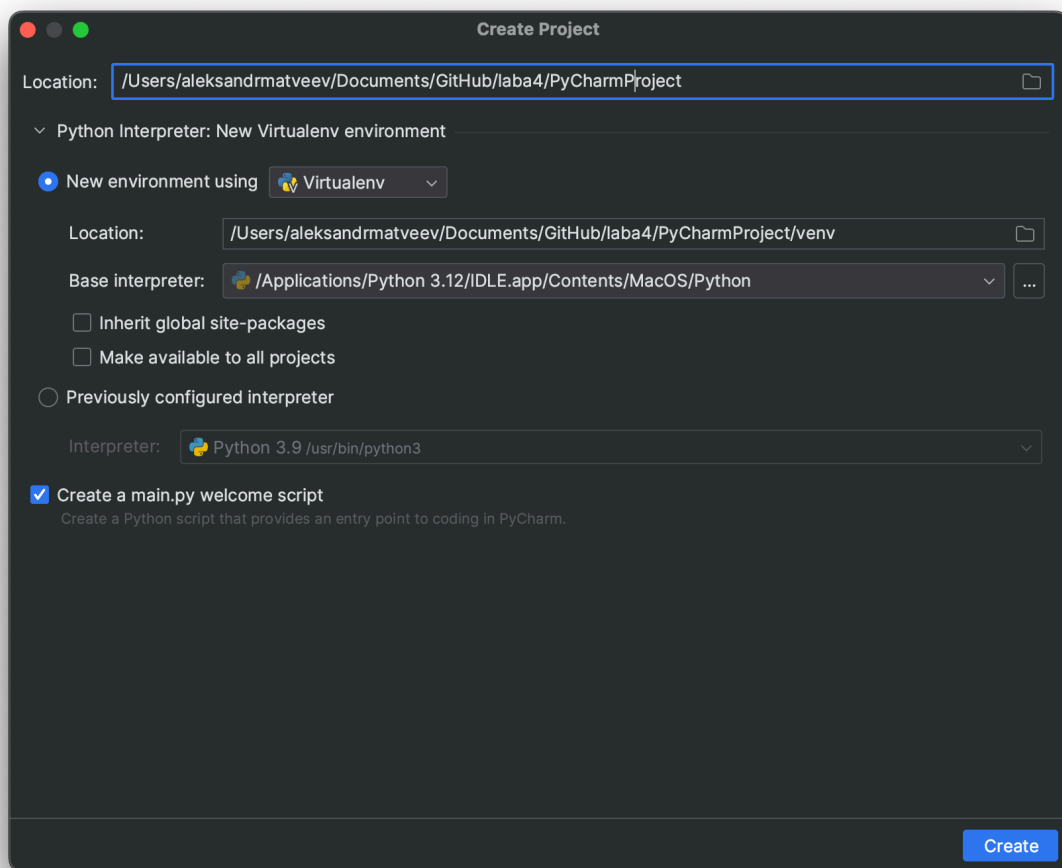


Рисунок 4 – Создание проекта

5. Решение задач 7-11.

```
user.py x
1 name = input("What is your name?")
2 age = input("How old are you?")
3 place = input("Where are you live?")
4 print(f"This is {name}\nIt is {age}\n(S)he live in {place}")
```

Рисунок 5 – Файл user.py

```
user.py arithmetic.py ×
1 print("Попробуйте решить задачу: (4 * 100 - 54)")
2 answer = int(input("Введите свой ответ: "))
3 print(f"Правильный ответ: {4 * 100 - 54}")
```

Рисунок 6 – Файл arithmetic.py

```
numbers.py × user.py arithmetic.py
1 first, second, third, fourth = [int(input()) for i in range(4)]
2 print(round((first + second) / (third + fourth), 2))
```

Рисунок 7 – Файл numbers.py

```
individual.py × numbers.py user.py arithmetic.py
1 # Вариант 11
2 # Известна стоимость 1 кг конфет, печенья и яблок. Найти стоимость всей покупки, если купили x кг конфет
3 candy_price, cookie_price, apple_price = (int(input("Введите стоимость конфет за кг: ")),
4                                             int(input("Введите стоимость печенья за кг: ")),
5                                             int(input("Введите стоимость яблок за кг: ")))
6 candy, cookie, apple = int(input("Введите вес купленных конфет: ")), int(
7     input("Введите вес купленного печенья: ")), int(input("Введите вес купленных яблок: "))
8 print(f"Стоимость всей покупки = {candy_price * candy + cookie_price * cookie + apple_price * apple}")
```

Рисунок 8 – Файл individual.py

```
1 #вариант 3
2 a3, a2, a1, b2, b1 = input(), input(), input(), input(), input(),
3 integer = str(int(a3 + a2 + a1) + int(b2 + b1))
4 print(integer[0], integer[1], integer[2])
```

Рисунок 9 – Файл individual.py

## 6. Выполнение коммита.

```
[aleksandrmatveev@MacBook-Air-Aleksandr laba4 % git commit -m "fix"
[develop b5c9d55] fix
6 files changed, 19 insertions(+)
create mode 100644 .DS_Store
create mode 100644 PyCharmProject/arithmetic.py
create mode 100644 PyCharmProject/individual.py
create mode 100644 PyCharmProject/individual_hard.py
create mode 100644 PyCharmProject/numbers.py
create mode 100644 PyCharmProject/user.py
```

Рисунок 10 – Коммит

7. Выполнение слияния ветки main с веткой develop.

```
[aleksandrmatveev@MacBook-Air-Aleksandr laba4 % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[aleksandrmatveev@MacBook-Air-Aleksandr laba4 % git merge develop
Merge made by the 'ort' strategy.
 .DS_Store | Bin 0 -> 6148 bytes
PyCharmProject/arithmetic.py | 3 +++
PyCharmProject/individual.py | 8 ++++++++
PyCharmProject/individual_hard.py | 2 ++
PyCharmProject/numbers.py | 2 ++
PyCharmProject/user.py | 4 ++++
6 files changed, 19 insertions(+)
create mode 100644 .DS_Store
create mode 100644 PyCharmProject/arithmetic.py
create mode 100644 PyCharmProject/individual.py
create mode 100644 PyCharmProject/individual_hard.py
create mode 100644 PyCharmProject/numbers.py
create mode 100644 PyCharmProject/user.py
[aleksandrmatveev@MacBook-Air-Aleksandr laba4 % git push
```

Рисунок 11 – Слияние ветки develop и main

### Контрольные вопросы:

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

Установка Python на Windows:

- 1) Перейти на сайт официальный сайт Python, скачать установочный файл.
- 2) Запустить загруженный установочный файл.
- 3) Выбрать способ установки.
- 4) Выбрать место установки.

Установка Python на Linux:

Для установки из репозитория в Ubuntu необходимо воспользоваться командой «`sudo apt-get install python3`».

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

Anaconda включает в себя не только сам Python, но и большое количество научных и аналитических пакетов, таких как NumPy, Pandas, Matplotlib, SciPy, а также удобную среду разработки, которая запускается в браузере.

3. Как осуществить проверку работоспособности пакета Anaconda?

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо ввести «`jupyter notebook`». В результате чего отобразиться процесс загрузки веб-среды Jupyter Notebook после чего запустится веб-сервер и среда разработки в браузере.

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

File-Settings-Project-Python Interpreter-в выпадающем меню указать путь до интерпретатора.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Нажать на кнопку Run, либо нажать сочетание клавиш: ctrl-shift-f10.

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

Интерактивный режим работы Python предоставляет возможность взаимодействия с интерпретатором языка непосредственно в реальном времени. Пакетный (или скриптовый) режим работы Python предполагает написание программы в файле с расширением .py, который затем выполняется целиком. В этом режиме все инструкции в программе выполняются последовательно, а результаты выводятся на экран либо сохраняются в файлы.

7. Почему язык программирования Python называется языком динамической типизации?

Потому что в нем типы переменных определяются автоматически во время выполнения программы, а не во время компиляции. Значит, что по ходу выполнения программного кода переменной можно менять ее тип.

8. Какие существуют основные типы в языке программирования Python?

К основным встроенным типам относятся:

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

1. int – целое число

2. float – число с плавающей точкой

3. complex – комплексное число

4. Списки (Sequence Type)

1. list – список

2. tuple – кортеж

3. range – диапазон



## 5. Строки (Text Sequence Type )

1. str

## 6. Бинарные списки (Binary Sequence Types)

1. bytes – байты
2. bytearray – массивы байт
3. memoryview – специальные объекты для доступа к внутренним данным объекта через protocol buffer

## 7. Множества (Set Types)

1. set – множество
2. frozenset – неизменяемое множество

## 8. Словари (Mapping Types)

1. dict – словарь.

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

При создании переменной (`a = 5`), в памяти создается объект типа `int` со значением 5, а переменная содержит ссылку на этот объект. Операция `=` лишь присваивает переменной ссылку на уже существующий объект.

10. Как получить список ключевых слов в Python?

Для этого нужно подключить модуль `keyword` и написать команду (`keyword.kwlist`).

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

Функция `id()` возвращает адрес объекта в памяти.

Функция `type()` возвращает тип объекта.

12. Что такое изменяемые и неизменяемые типы в Python?

Изменяемые типы дают возможность переписывать значения в памяти после создания, а неизменяемые типы не могут быть изменены после создания.

13. Чем отличаются операции деления и целочисленного деления?

Python операция деления (/) выполняет обычное деление чисел с плавающей точкой и возвращает результат типа float. Целочисленное деление (//) выполняет деление и возвращает целую часть результата типа int, отбрасывая дробную часть.

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

Для создания комплексного числа можно использовать функцию complex(a, b), либо записать число в виде a + bj. В python их можно вычитать, складывать, делить, умножать и возводить в степень.

15. Каково назначение и основные функции библиотеки (модуля) math?

Библиотека math в Python предоставляет функции для выполнения математических операций и обработки чисел. Основные функции для обработки чисел: math.sqrt(), math.pow(), math.exp(), math.log(), math.sin(), math.cos(), math.tan(). Тригонометрические функции: math.sin(), math.cos(), math.tan(), math.asin(), math.acos(), math.atan() и многие другие.

16. Каково назначение именованных параметров sep и end в функции print()?

Через параметр (sep) можно указать разделитель строк. Параметр (end) дает возможность изменить, какой символ будет выведен в конце строки, по умолчанию перенос на новую (\n).

17. Каково назначение метода format()? Какие еще существуют средства для форматирования строк в Python?

Метод format() в Python служит для форматирования строк. Он позволяет вставлять значения переменных в строки, указывая их место с помощью фигурных скобок '{}'. Также для форматирования строк в Python есть инструмент под названием: f-строки.

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

Для ввода значений целочисленной и вещественной переменной в языке Python можно использовать функции `float(input())` или `int(input())`.