

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 «Программирование на Python»**

Выполнил:  
Гайчук Дарья Дмитриевна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика  
и вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А.-доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** «Основы ветвления Git»

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

**Порядок выполнения работы:**

1. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

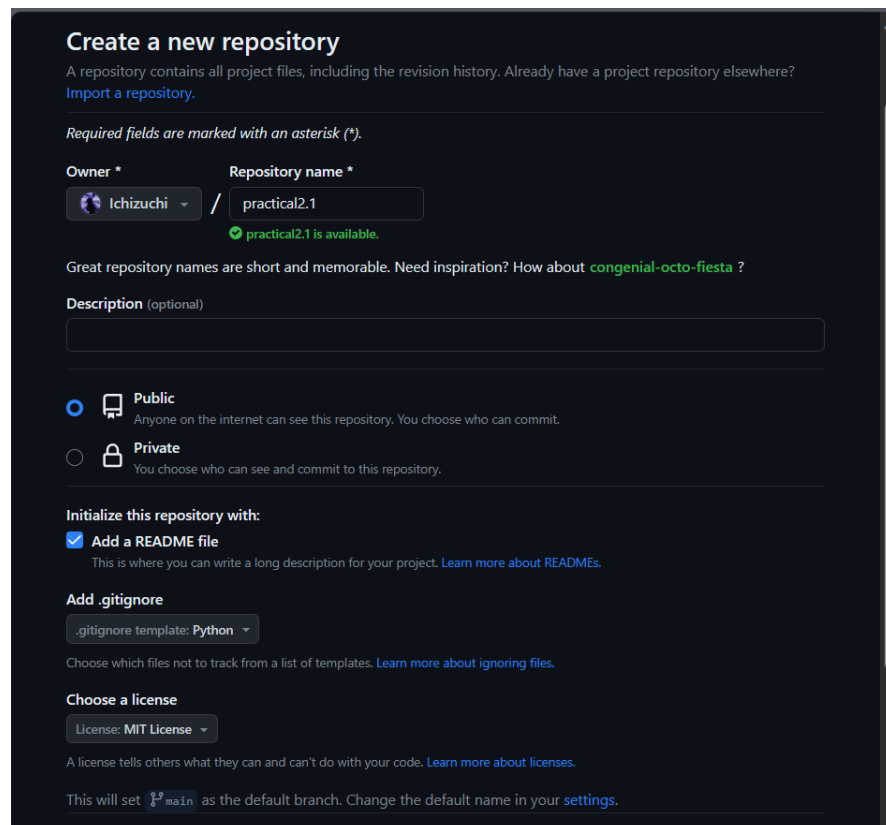


Рисунок 1. Новый репозиторий

2. Выполнила клонирование репозитория.

3. Использовала такую модель ветвления как git-flow.

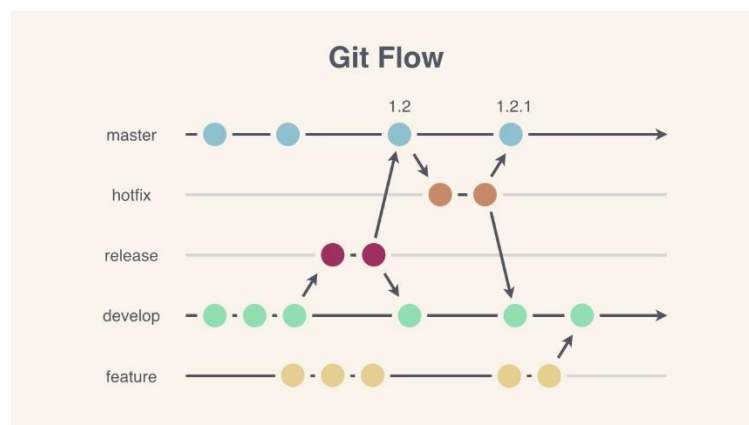
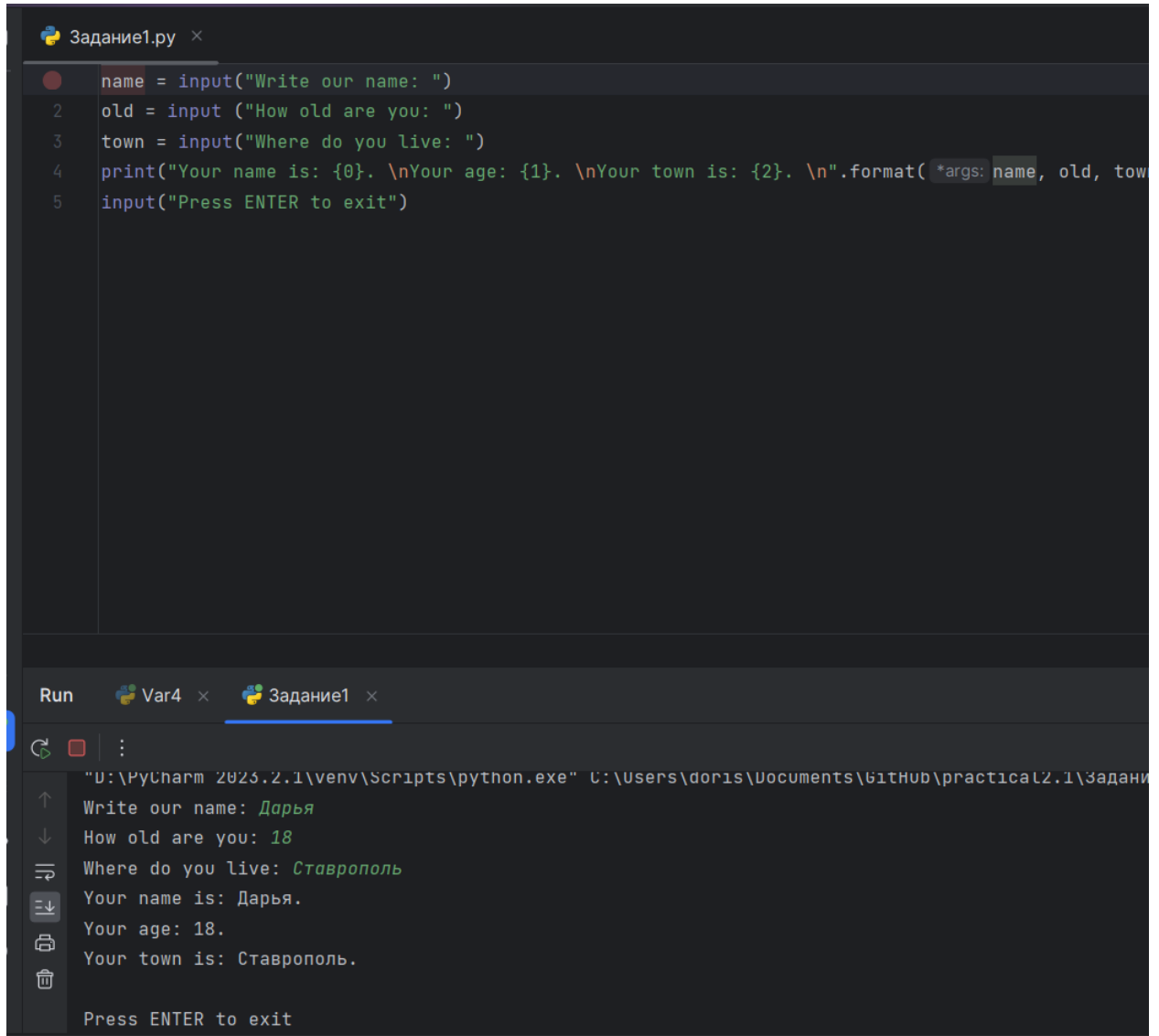


Рисунок 3. Модель ветвления git-flow.

4. Написала программу и проверила на работоспособность в файле user.py. Необходимо, чтобы данная программа запрашивала у пользователя его имя, возраст, где он живет и выводила эти данные, написанные пользователем на экран.



The screenshot displays the PyCharm IDE interface. The top editor window, titled 'Задание1.py', contains the following Python code:

```
1 name = input("Write our name: ")
2 old = input("How old are you: ")
3 town = input("Where do you live: ")
4 print("Your name is: {0}. \nYour age: {1}. \nYour town is: {2}. \n".format(*args: name, old, town))
5 input("Press ENTER to exit")
```

The bottom panel shows the 'Run' output window for 'Задание1'. It displays the execution path and the user's input/output:

```
"D:\PyCharm 2023.2.1\venv\Scripts\python.exe" C:\Users\doris\Documents\GitHub\practical2.1\Задание1.py
Write our name: Дарья
How old are you: 18
Where do you live: Ставрополь
Your name is: Дарья.
Your age: 18.
Your town is: Ставрополь.
Press ENTER to exit
```

Рисунок 4. Программа и ее результат

5. Создала новый файл Задание2.py и написала программу в develop ветке. Необходимо, чтобы данная программа предлагала бы пользователю решить пример  $4 * 100 - 54$ . Потом выводила бы на экран правильный ответ и ответ пользователя.

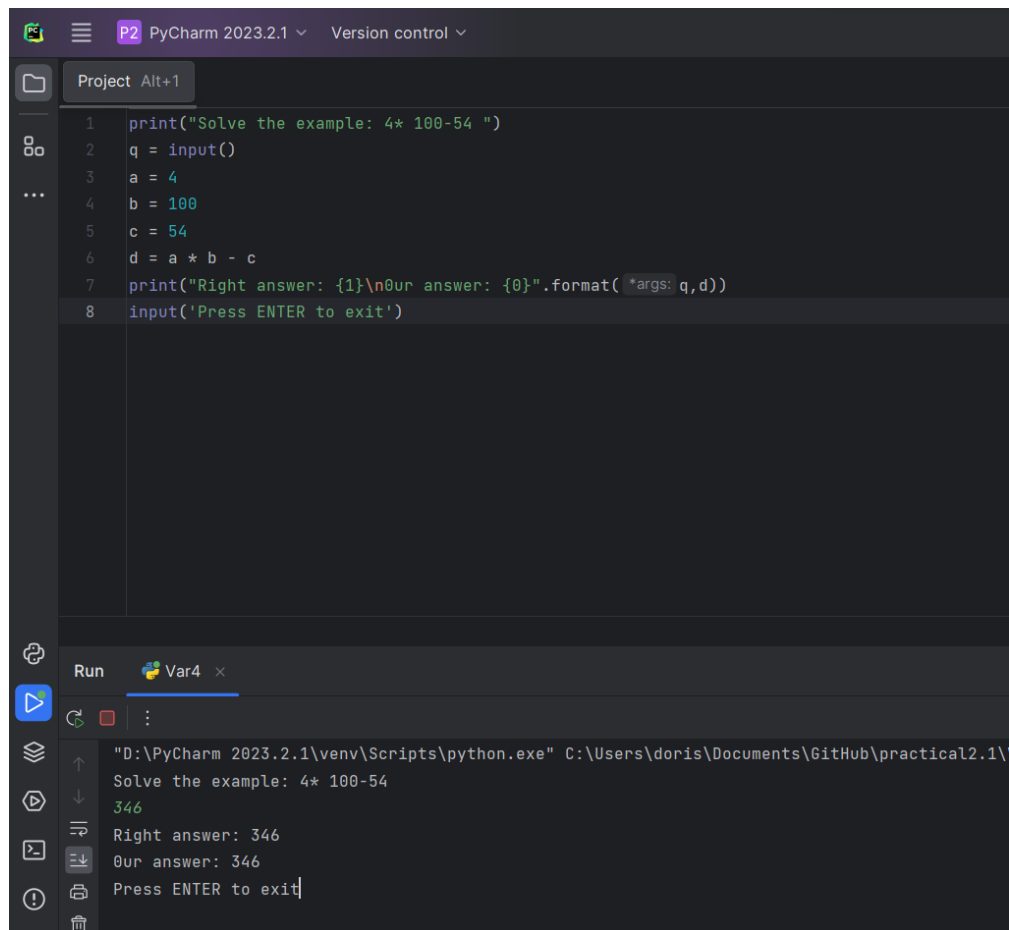
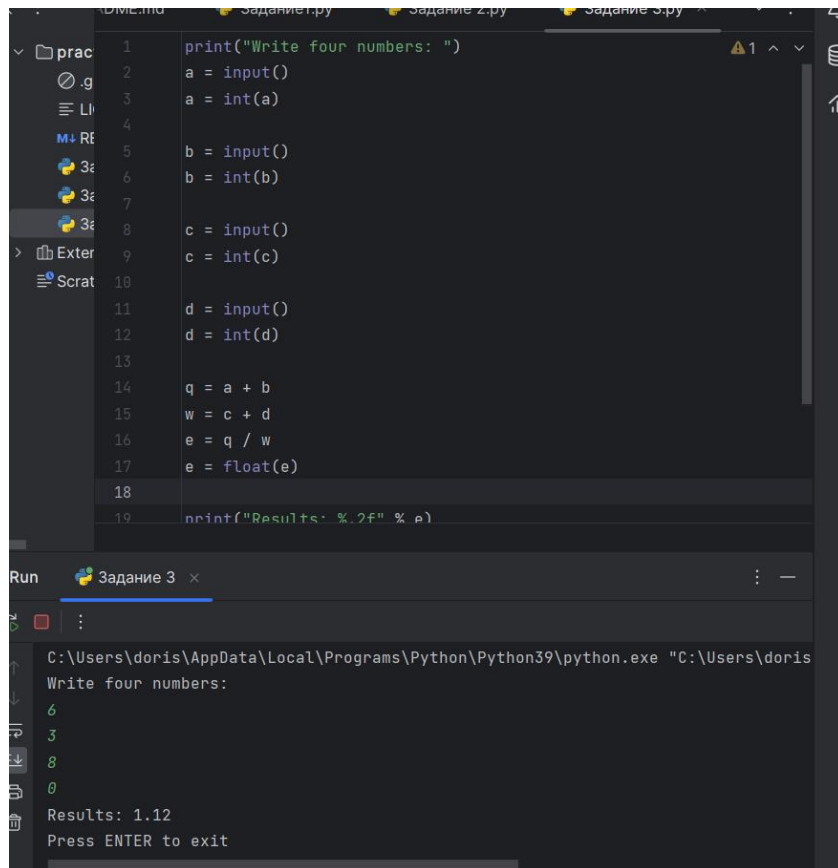


Рисунок 5. Программа и ее результат.

6. Создала новый файл Задание3.py Написала программу и проверила на работоспособность в файле Задание3.py. Необходимо, чтобы данная программа запросила у пользователя четыре числа, далее отдельно складывала первые два числа и вторые два числа, разделила первую сумму на вторую и вывела результат на экран (ответ должен содержать 2 цифры после запятой).



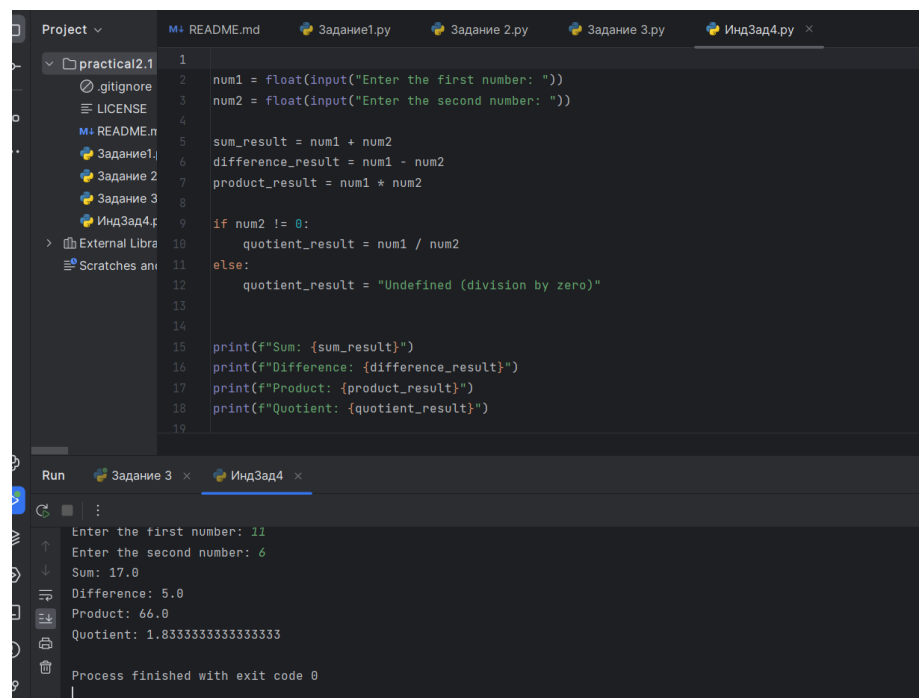
```
1 print("Write four numbers: ")
2 a = input()
3 a = int(a)
4
5 b = input()
6 b = int(b)
7
8 c = input()
9 c = int(c)
10
11 d = input()
12 d = int(d)
13
14 q = a + b
15 w = c + d
16 e = q / w
17 e = float(e)
18
19 print("Results: %.2f" % e)
```

Run Задание 3

```
C:\Users\doris\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\doris\
Write four numbers:
6
3
8
0
Results: 1.12
Press ENTER to exit
```

Рисунок 6. Программа и ее результат.

7. Сделала индивидуальное в соответствии со списком задание в ветке develop. Вариант 4. Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.



```
1
2 num1 = float(input("Enter the first number: "))
3 num2 = float(input("Enter the second number: "))
4
5 sum_result = num1 + num2
6 difference_result = num1 - num2
7 product_result = num1 * num2
8
9 if num2 != 0:
10     quotient_result = num1 / num2
11 else:
12     quotient_result = "Undefined (division by zero)"
13
14
15 print(f"Sum: {sum_result}")
16 print(f"Difference: {difference_result}")
17 print(f"Product: {product_result}")
18 print(f"Quotient: {quotient_result}")
19
```

Run Задание 3 ИндЗад4

```
Enter the first number: 11
Enter the second number: 6
Sum: 17.0
Difference: 5.0
Product: 66.0
Quotient: 1.8333333333333333
Process finished with exit code 0
```

Рисунок 7. Программа индивидуального задания и ее результат.

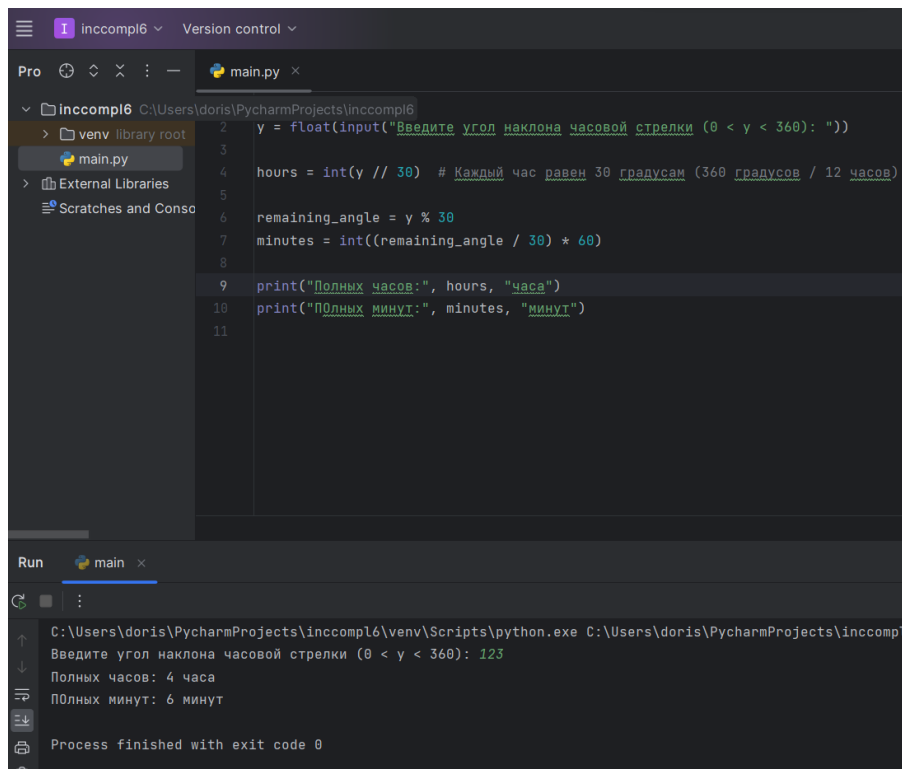
8. Слила ветку develop с веткой main, Отправка на гитхаб.

```
● @Ichizuchi →/workspaces/practical2.1 (develop) $ git checkout main
branch 'main' set up to track 'origin/main'.
Switched to a new branch 'main'
● @Ichizuchi →/workspaces/practical2.1 (main) $ git merge develop
Updating 7404b3d..33d5399
Fast-forward
 ...20\260\320\275\320\270\320\265 2.py" | 8 +++++
 ...20\260\320\275\320\270\320\265 3.py" | 20 ++++++
 ...320\260\320\275\320\270\320\2651.py" | 5 +++
 ...320\264\320\227\320\260\320\2644.py" | 18 ++++++
4 files changed, 51 insertions(+)
create mode 100644 "\320\227\320\260\320\264\320\260\320\275\320\270\320\265 2.py"
create mode 100644 "\320\227\320\260\320\264\320\260\320\275\320\270\320\265 3.py"
create mode 100644 "\320\227\320\260\320\264\320\260\320\275\320\270\320\2651.py"
create mode 100644 "\320\230\320\275\320\264\320\227\320\260\320\264 4.py"
● @Ichizuchi →/workspaces/practical2.1 (main) $ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ichizuchi/practical2.1
7404b3d..33d5399 main -> main
○ @Ichizuchi →/workspaces/practical2.1 (main) $
```

Рисунок 8. Слияние веток.

Задача повышенной сложности

6. С начала суток часовая стрелка повернулась на градусов ( $0 < y < 360$ ,  $y$ - вещественное число). Определить число полных часов и число полных минут, прошедших с начала суток.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations and a 'Run' button. The main editor displays a file named 'main.py' with the following Python code:

```
1 y = float(input("Введите угол наклона часовой стрелки (0 < y < 360): "))
2
3
4 hours = int(y // 30) # Каждый час равен 30 градусам (360 градусов / 12 часов)
5
6 remaining_angle = y % 30
7 minutes = int((remaining_angle / 30) * 60)
8
9 print("Полных часов:", hours, "часа")
10 print("Полных минут:", minutes, "минут")
11
```

The bottom panel shows the 'Run' output for 'main.py'. The execution path is 'C:\Users\doris\PycharmProjects\inccompl6\venv\Scripts\python.exe'. The output shows the user input '123' and the resulting calculations: 'Полных часов: 4 часа' and 'Полных минут: 6 минут'. The process finished with exit code 0.

Рисунок 9. Программа и ее выполнение

## Контрольные вопросы

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

Для начала скачиваем дистрибутив по ссылке. Запускаем скачанный файл. Выбираем способ установки (install now или customize installation). Отмечаем необходимые опции для установки. Выбираем место установки (доступно при выборе Customize installation).

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

Anaconda наиболее известна как дистрибутив Python со встроенным в него пакетным менеджером conda. Она позволяет изолировать окружение проекта от системной версии Python, который критически необходим для работы системы. Использование `sudo pip` считается плохой практикой. Также conda позволяет без проблем переносить окружение с одной машины на другую. Кроме того, если вы что-то сломаете, то с Anaconda вы всегда сможете откатиться на более старую версию окружения.

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 отобразится процесс загрузки веб-среды Jupyter Notebook

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

Запускаем программу. И он не содержит ни одного файла с текстом программы. Создадим его. Нажимаем правую кнопку мыши, выбираем «New» -> «Python File». Вводим имя файла, например, `ex1` и этот файл автоматически добавляется в наш проект. Здесь мы можем писать наши программы, на Питоне.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Здесь при первом запуске необходимо создать новый проект. Нажимаем «Create New Project». В поле «Location» указывается расположение проекта и его имя.

6. В чем суть интерактивного и пакетного режимов работы Python? Его суть в удобстве использования языка Питон, а именно его внешний вид и простота, сопровождаемая автоматикой действий, позволяющей пользователям работать с данными программами.

7. Почему язык программирования Python называется языком динамической типизации?

Потому что в Питоне тип переменной определяется непосредственно при выполнении программы.

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)

- int – целое число
- float – число с плавающей точкой
- complex – комплексное число

### 4) Списки (Sequence Type)

- list – список
- tuple – кортеж
- range – диапазон
- Строки (Text Sequence Type)
- str Бинарные списки (Binary Sequence Types)
- bytes – байты
- bytearray – массивы байт
- memoryview – специальные объекты для доступа к внутренним

данным

### 5) Объекты через protocol buffer Множества (Set Types)

- set – множество
- frozenset – неизменяемое множество
- Словари (Mapping Types)
- dict – словарь

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

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Например, строка: `b=5` Целочисленное значение 5 в рамках языка Python по сути своей является объектом. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными следует понимать как непосредственно сами объекты, так и отношения между ними. Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга,

а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор.

#### 10. Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

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

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`. Тип переменной можно определить с помощью функции `type()`.

#### 12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (immutable) 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 (mutable) типам относятся: списки (`list`), множества (`set`), словари (`dict`). Как уже было сказано ранее, при создании переменной, вначале создается объект, который имеет уникальный идентификатор, тип и значение, после этого переменная может ссылаться на созданный объект.

#### 13. Чем отличаются операции деления и целочисленного деления?

Целочисленное деление (`div`) отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Перед выполнением операции оба операнда округляются до целых значений.

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

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде `a + bj`.

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

По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`. В стандартную поставку Python входит библиотека `math`, в которой содержится большое количество часто используемых математических функций. Для работы с данным модулем его предварительно нужно импортировать.

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

`sep` – с помощью этого параметра вы можете указать разделитель строк. А по умолчанию в качестве разделителя используется пробел. `end` – этот параметр позволяет указать, что нужно добавить после последней строки. По умолчанию добавляется управляющий символ `'\n'` (перевод строки)

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()`, который определен у строк, позволяет форматировать строку, вставляя в нее на место плейсхолдеров определенные значения. Для вставки в строку используются специальные параметры, которые обрамляются фигурными скобками (`{}`).

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

Если необходимо ввести целочисленное значение можно использовать код: `A = input()` `A = int()` Если необходимо ввести вещественное значение можно использовать код: `A = input()` `A = float()` Для ввода нужно нажать Enter после завершения набора текста. Обычно Enter добавляет символ новой строки (`\n`), но не в этом случае. Введенная строка просто будет передана приложению.

Вывод: исследовала процесс установки и базовые возможности языка Python версии 3.x.