

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

Отчет о лабораторной работе №2.1 по дисциплине основы программной инженерии

Выполнил:
Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

Выполнение:

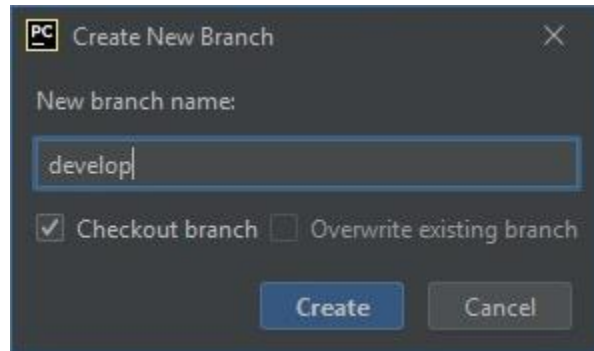


Рисунок 2.1 – Создание ветки develop

```
name = input("What is your name?\n")
age = input("How old are you?\n")
residence = input("Where are you live?\n")
print(f"This is {name}")
print(f"It is {age}")
print(f"(S)he live in {residence}")
```

Рисунок 2.2 – Программа user.py

```
What is your name?
Vladimir
How old are you?
18
Where are you live?
Stavropol
This is Vladimir
It is 18
(S)he live in Stavropol
```

Рисунок 2.3 – Пример работы программы user.py

```
ans = input("4 * 100 - 54 = ")
print(f"Your answer: {ans}")
print(f"Correct answer {4 * 100 - 54}")
```

Рисунок 2.4 – Программа arithmetic.py

```
4 * 100 - 54 = 346
Your answer: 346
Correct answer 346
```

Рисунок 2.5 – Пример работы программы arithmetic.py

```

num1 = int(input("Enter first number = "))
num2 = int(input("Enter second number = "))
num3 = int(input("Enter third number = "))
num4 = int(input("Enter fourth number = "))
sum1 = num1 + num2
sum2 = num3 + num4
result = sum1 / sum2
print("%.2f" % result)

```

Рисунок 2.6 – Программа numbers.py

```

Enter first number = 0
Enter second number = 2
Enter third number = 0
Enter fourth number = 4
0.50

```

Рисунок 2.7 – Пример работы программы numbers.py

Индивидуальное задание, вариант номер 1.

```

# Variant 1
from math import fabs, sqrt

num1 = int(input("Enter first number = "))
num2 = int(input("Enter second number = "))
arithmetic = (fabs(num1) + fabs(num2)) / 2
geometric = sqrt(fabs(num1) * fabs(num2))
print(f"Arithmetical mean = {arithmetic}")
print("Geometric mean = %.3f" % geometric)

```

Рисунок 2.8 – Программа individual.py

```

Enter first number = 4
Enter second number = 4
Arithmetical mean = 4.0
Geometric mean = 4.000

```

Рисунок 2.9 – Пример работы программы individual.py

Задача повышенной сложности, вариант номер 5.

```
# Variant 5

h = int(input("Hours = "))
m = int(input("Minutes = "))
s = int(input("Seconds = "))
one_hour = 30
one_min = one_hour / 60
one_sec = one_min / 60
deg = h * one_hour + m * one_min + s * one_sec
print("Degrees = %.2f" % deg)
```

Рисунок 2.10 – Программа hard.py

```
Hours = 6
Minutes = 30
Seconds = 50
Degrees = 195.42
```

Рисунок 2.11 – Пример работы программы hard.py

```
F:\pythonProject\laboratory-4>git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arithmetic.py
    hard.py
    individual.py
    numbers.py
    user.py

nothing added to commit but untracked files present (use "git add" to track)
```

Рисунок 2.12 – Статус файлов

```
F:\pythonProject\laboratory-4>git add .

F:\pythonProject\laboratory-4>git commit -m "Add new programs"
[develop b43b0d7] Add new programs
5 files changed, 36 insertions(+)
create mode 100644 arithmetic.py
create mode 100644 hard.py
create mode 100644 individual.py
create mode 100644 numbers.py
create mode 100644 user.py
```

Рисунок 2.13 – Коммит файлов

```
F:\pythonProject\laboratory-4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

F:\pythonProject\laboratory-4>git merge develop
Updating c1f85a3..b43b0d7
Fast-forward
 arithmetic.py | 3 +++
 hard.py       | 10 ++++++++
 individual.py | 9 ++++++++
 numbers.py    | 8 ++++++++
 user.py       | 6 +++++
 5 files changed, 36 insertions(+)
 create mode 100644 arithmetic.py
 create mode 100644 hard.py
 create mode 100644 individual.py
 create mode 100644 numbers.py
 create mode 100644 user.py
```

Рисунок 2.14 – Merge двух ветки develop в main

```
F:\pythonProject\laboratory-4>git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 6 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.10 KiB | 1.10 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/HAXF13D/laboratory-4
 c1f85a3..b43b0d7 main -> main
```

Рисунок 2.14 – Отправка изменений на сервер

Ссылка на репозиторий: <https://github.com/HAXF13D/laboratory-4>

ОТВЕТЫ НА ВОПРОСЫ:

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

Windows:

1. Запустите скачанный установочный файл.
2. Выберет способ установки.
3. Отметьте необходимые опций установки (доступно при выборе Customize installation)
4. Выберете место установки (доступно при выборе Customize installation)
5. После успешной установки вас ждет следующее сообщение.

Linux:

sudo apt-get install python3

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

Он включает в себя набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

3. Как осуществить проверку работоспособности пакета Anaconda?

Пуск -> Anaconda3 (64-bit) -> Anaconda Prompt. В появившейся командной строке необходимо ввести: jupyter notebook

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

Во время создания нового проекта выбрать директорию установки python.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Добавить файл main.py и написать простую программу, print("Hello world"). В правом верхнем углу выбрать исполняемый файл и нажать на зеленую стрелочку.

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

Интерактивный режим позволяет использовать python, как калькулятор, не запоминая код программы в файл. Пакетный же режим исполняет код написанный в файле.

7. Почему язык программирования Python называется языком динамической типизации?

Потому что при объявлении переменных не указывается их тип данных.

8. Какие существуют основные типы в языке программирования Python?

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?

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 (об этом чуть позже). 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.

При инициализации переменной, на уровне интерпретатора, происходит следующее:

- создается целочисленный объект 5 (можно представить, что в этот момент создается ячейка
- и 5 кладется в эту ячейку);
- данный объект имеет некоторый идентификатор, значение: 5, и тип: целое число;
- посредством оператора “=” создается ссылка между переменной b и целочисленным объектом 5 (переменная b ссылается на объект 5).

10. Как получить список ключевых слов в Python?

```
import keyword  
>>> print("Python keywords: ", 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`) неизменяемые множества (`frozen set`).

К изменяемым (mutable) типам относятся: списки (`list`), множества (`set`), словари (`dict`).

13. Чем отличаются операции деления и целочисленного деления?

Целочисленное деление возвращает целую часть от деления, тип данных `int`, деление возвращает `float`

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

`complex(a, b)`, `a + bj`. `z = a + bj`

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

входит библиотека `math`, в которой содержится большое количество часто используемых математических функций.

`math.ceil(x)`

`math.fabs(x)`

`math.factorial(x)`

`math.floor(x)`

`math.exp(x)`

`math.log2(x)`

`math.log10(x)`

`math.log(x[, base])`

`math.pow(x, y)`

`math.sqrt(x)`

`math.cos(x)`

`math.sin(x)`

`math.pi`

`math.e`

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

`sep` – разделитель между данными

`end` – символ в конце 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.

Форматирование по стандартам C – `print("%f" % (2.3))`

Метод `.format` – `print("{0}".format(2))`

f строки – `print(f"{2}")`

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

Целое число – `int(input())`

Вещественное – `float(input())`