

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 №24**  
**дисциплины «Основы программной инженерии»**

Выполнил:  
Матвеев Александр Иванович  
2 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, 2024 г.

**Тема:** Элементы объектно-ориентированного программирования в языке Python.

**Цель работы:** приобретение навыков по работе с классами и объектами при написании программ с помощью языка программирования 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 \*** laba10

✔ laba10 is available.

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

**Description** (optional)

☒ **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.](#)

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

**Create repository**

Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
[→ GitHub git clone https://github.com/SashkaHacker/laba16.git
Cloning into 'laba16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.

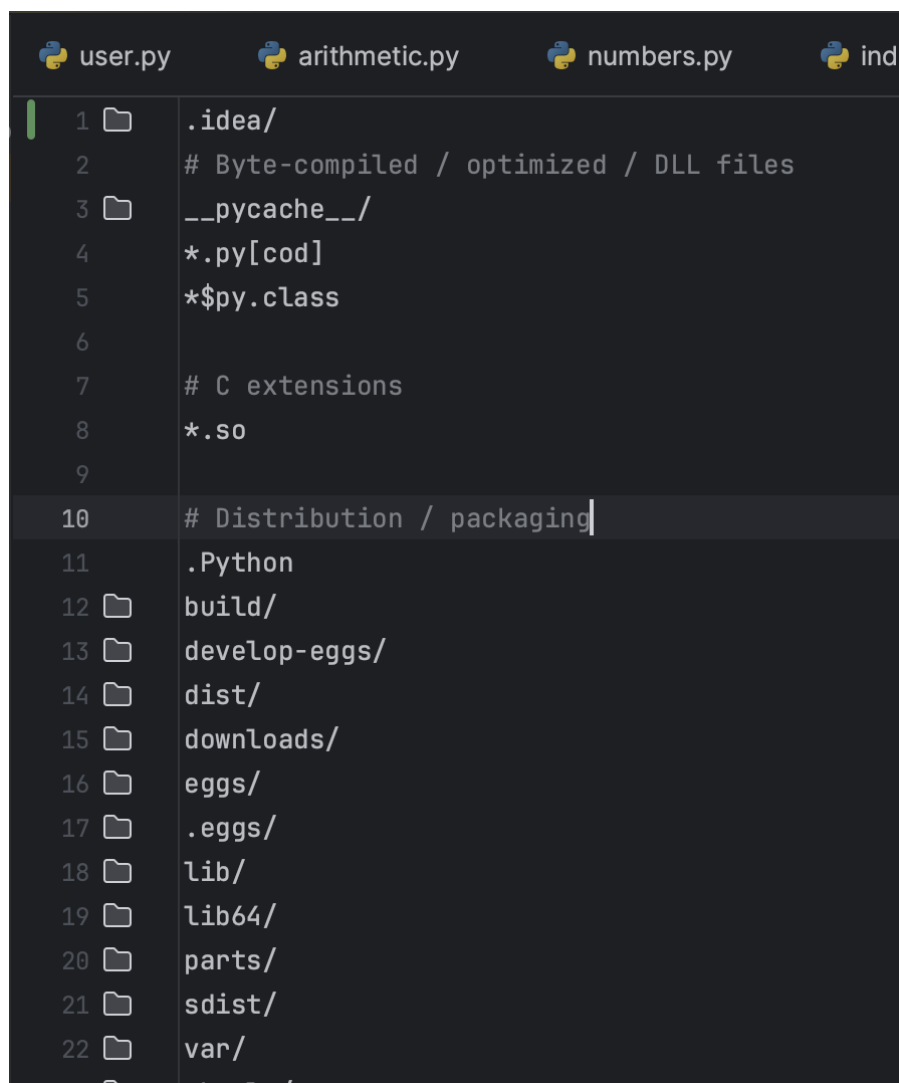
A screenshot of a code editor with a dark theme. At the top, there are four tabs: 'user.py', 'arithmetic.py', 'numbers.py', and 'ind'. The 'user.py' tab is active. The editor shows the content of a '.gitignore' file. The text is as follows:  
1 | .idea/  
2 | # Byte-compiled / optimized / DLL files  
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/  
The cursor is positioned at the end of line 10, after the text '# Distribution / packaging'.

Рисунок 3 – Файл .gitignore

#### 4. Проработка примера №1 из лабораторной работы.

```
C:\Users\Sashka\Documents\GitHub\laba24\env\Scripts\python.exe C:\Users\Sashka\Documents\GitHub\laba24\examples\ex1.py
3/4
Введите обыкновенную дробь: 5/6
5/6
19/12
1/12
5/8
10/9
```

Рисунок 4 – Демонстрация работы примера №1

#### 5. Выполнение индивидуального задания №1 (Вариант - 10).

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Вариант 10

class LinearEquation:
    def __init__(self, first, second):
        if not isinstance(first, (int, float)) or not isinstance(second, (
            int, float)):
            raise ValueError("Коэффициенты должны быть числами.")
        self.first = first
        self.second = second

    def read(self):
        self.first = float(input("Введите коэффициент a: "))
        self.second = float(input("Введите коэффициент b: "))

    def display(self):
        print(f"Линейное уравнение: {self.first}x + {self.second} = 0")

    def function(self, x):
        return self.first * x + self.second

if __name__ == '__main__':
    equation1 = LinearEquation(first=1, second=2)
    equation1.display()
    print(equation1.function(10))

    equation2 = LinearEquation(first=0, second=0)
    equation2.read()
    equation2.display()
    x = float(input("Введите значение x для вычисления функции: "))
    print(f"Значение функции: {equation2.function(x)}")
```

Рисунок 5 – Код программы

```
Линейное уравнение:  $1x + 2 = 0$   
12  
Введите коэффициент а: 1  
Введите коэффициент b: 2  
Линейное уравнение:  $1.0x + 2.0 = 0$   
Введите значение x для вычисления функции: 10  
Значение функции: 12.0
```

Рисунок 6 – Демонстрация работы программы

## 6. Выполнение индивидуального задания №2 (Вариант - 10).

```
Сумма денег состоит из:  
5000 руб. - 1 шт.  
1000 руб. - 1 шт.  
100 руб. - 1 шт.  
50 руб. - 1 шт.  
10 руб. - 1 шт.  
5 руб. - 1 шт.  
0.5 руб. - 1 шт.  
0.1 руб. - 1 шт.  
Общая стоимость: 6165,6 руб.  
  
Сумма денег состоит из:  
5000 руб. - 1 шт.  
100 руб. - 1 шт.  
10 руб. - 1 шт.  
0.5 руб. - 1 шт.  
Общая стоимость: 4055,4 руб.  
  
Сумма денег состоит из:  
5000 руб. - 4 шт.  
100 руб. - 4 шт.  
10 руб. - 4 шт.  
0.5 руб. - 4 шт.  
Общая стоимость: 20442,0 руб.  
  
Сумма денег состоит из:  
5000 руб. - 0.2 шт.  
100 руб. - 0.2 шт.  
10 руб. - 0.2 шт.  
0.5 руб. - 0.2 шт.  
Общая стоимость: 1022,1 руб.
```

Рисунок 7 – Демонстрация работы программы

### Контрольные вопросы:

1. Класс в Python объявляется с помощью ключевого слова `class`, за которым следует имя класса и двоеточие. Тело класса содержит методы и атрибуты класса.
2. Атрибуты класса являются общими для всех экземпляров класса, в то время как атрибуты экземпляра уникальны для каждого созданного объекта класса.
3. Методы класса предназначены для выполнения действий, связанных с объектами класса или самим классом. Они могут взаимодействовать с атрибутами объекта и выполнять различные операции.
4. Метод `__init__()` предназначен для инициализации нового экземпляра класса. Это конструктор класса, который вызывается при создании объекта и может принимать аргументы для инициализации атрибутов объекта.
5. `self` представляет экземпляр класса и используется для доступа к атрибутам и методам объекта из его методов.
6. Атрибуты в класс добавляются путем объявления их в методе `__init__()` с использованием `self`, например, `self.attribute_name = value`, или как переменные класса, объявленные в теле класса.
7. Управление доступом к методам и атрибутам в Python осуществляется с помощью специальных префиксов (одно или два подчеркивания) перед именами атрибутов или методов. Однако Python не поддерживает строгое приватное или защищенное управление доступом, как в некоторых других языках.
8. Функция `isinstance` используется для проверки, является ли объект экземпляром указанного класса или класса, являющегося подклассом указанного класса. Это позволяет проверять тип объекта во время выполнения.