

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 №16**  
**дисциплины «Программирование на Python»**

Выполнил:  
Касимов Асхаб Арсенович  
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., канд. технических  
наук, доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

Тема: Модули и пакеты

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.

2. Дополнил файл .gitignore необходимыми правилами.

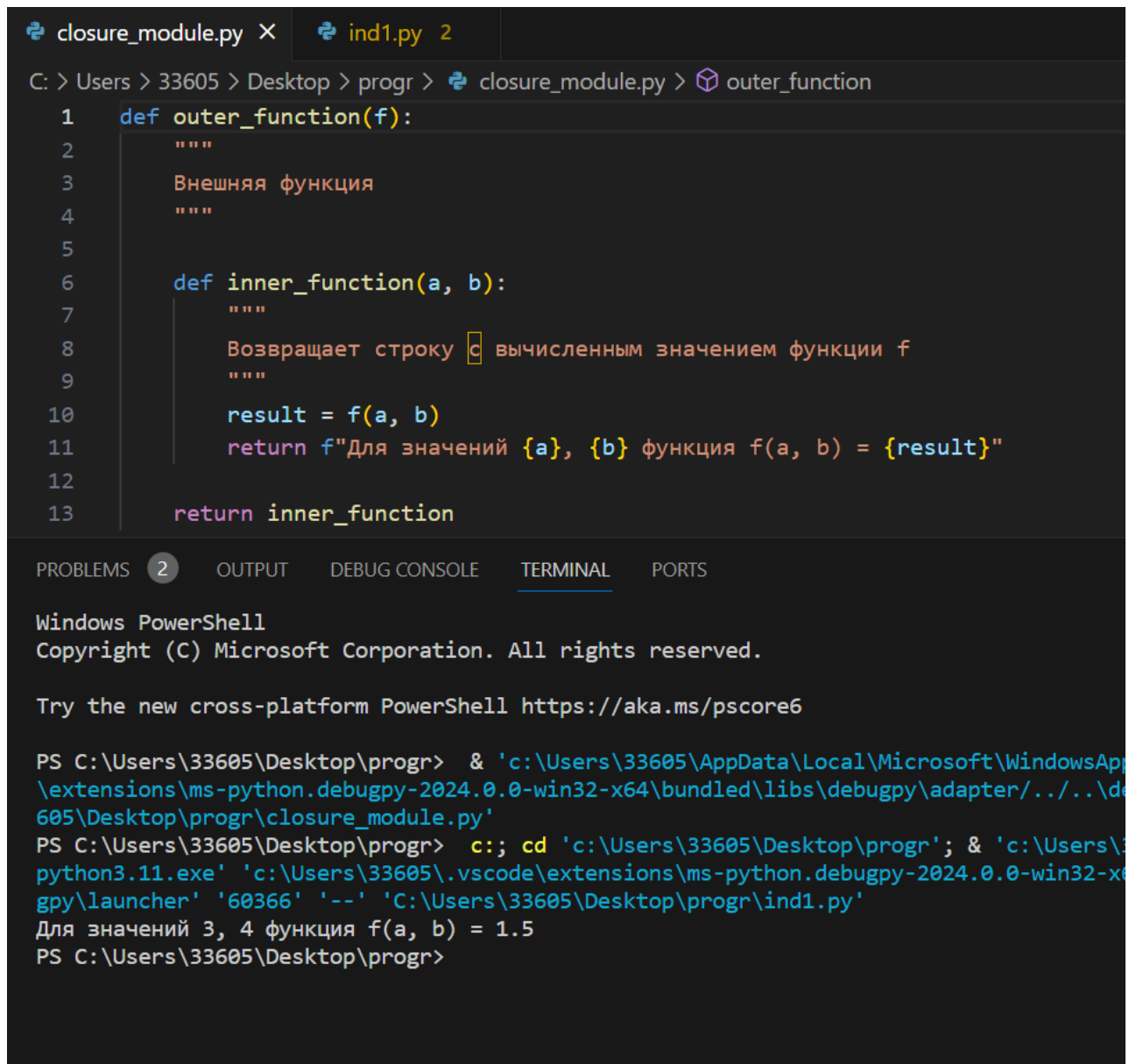
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.

4. Выполнил индивидуальное задание 1. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`.

Индивидуальное задание лабораторной работы 2.11 8 варианта: Используя замыкания функций, объявите внутреннюю функцию, которая принимает два параметра  $a$ ,  $b$ , а затем, возвращает строку в формате: «Для значений  $a$ ,  $b$  функция  $f(a,b) = \langle \text{число} \rangle$ » где число – это вычисленное значение функции  $f$ . Ссылка на  $f$  передается как аргумент внешней функции. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы. Функцию  $f$  придумайте самостоятельно (она должна что то делать с двумя параметрами  $a$ ,  $b$  и возвращать результат).

```
closure_module.py X
C: > Users > 33605 > Desktop > progr > closure_module.py > outer_function
1 def outer_function(f):
2     """
3     Внешняя функция
4     """
5
6     def inner_function(a, b):
7         """
8         Возвращает строку с вычисленным значением функции f
9         """
10        result = f(a, b)
11        return f"Для значений {a}, {b} функция f(a, b) = {result}"
12
13    return inner_function
14
15
16 def f(a, b):
17     """
18     Делит параметры друг на друга и умножает на 2
19     """
20    return a / b * 2
21
```

Рисунок 1. Модуль для индивидуального задания 1



The image shows a Visual Studio Code editor window with two tabs: `closure_module.py` and `ind1.py 2`. The active tab is `closure_module.py`, which contains the following Python code:

```
1 def outer_function(f):
2     """
3     Внешняя функция
4     """
5
6     def inner_function(a, b):
7         """
8         Возвращает строку c вычисленным значением функции f
9         """
10        result = f(a, b)
11        return f"Для значений {a}, {b} функция f(a, b) = {result}"
12
13    return inner_function
```

Below the editor, the **TERMINAL** panel is open, showing the execution of the script in a Windows PowerShell environment. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\33605\Desktop\progr> & 'c:\Users\33605\AppData\Local\Microsoft\WindowsApp
\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter/../../de
605\Desktop\progr\closure_module.py'
PS C:\Users\33605\Desktop\progr> c.;; cd 'c:\Users\33605\Desktop\progr'; & 'c:\Users\3
python3.11.exe' 'c:\Users\33605\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x6
gpy\launcher' '60366' '--' 'C:\Users\33605\Desktop\progr\ind1.py'
Для значений 3, 4 функция f(a, b) = 1.5
PS C:\Users\33605\Desktop\progr>
```

Рисунок 2. Результат работы программы из индивидуального задания 1

4. Выполнил индивидуальное задание 2. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import` . Настроить соответствующим образом переменную `__all__` в файле `__init__.py` пакета.

Индивидуальное задание лабораторной работы 2.11 8 варианта: Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

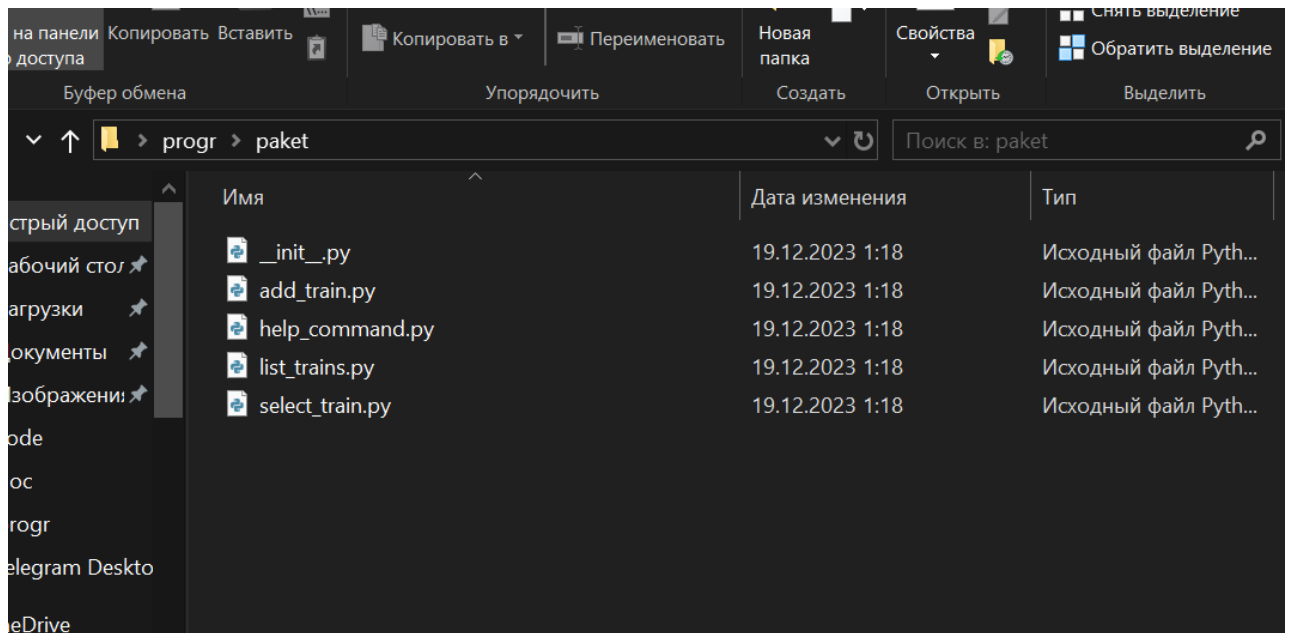


Рисунок 3. Созданный пакет

```
1  __all__ = ['add_train', 'help_command', 'list_trains', 'select_train']
```

Рисунок 4. Настройка переменной `__all__` в файле `__init__.py` пакета

```
ind2.py 6
C: > Users > 33605 > Desktop > progr > ind2.py > ...
1 py#!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 from paket import *
6 import sys
7
8
9 def main():
10     """
11     Главная функция программы.
12     """
13     trains = []

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS Python Debug Console + - [] ..
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Номер поезда: 777
Название пункта назначения: Moscow
Время отправления: 12:00
>>> list
+-----+-----+-----+-----+
| № |      Номер поезда      | Пункт назначения | Время |
+-----+-----+-----+-----+
|  1 | 777                    | Moscow          | 12:00 |
+-----+-----+-----+-----+
>>> select

Введите номер поезда для поиска: 777
{'номер поезда': '777', 'название пункта назначения': 'Moscow', 'время отправления': '12:00'}
```

Рисунок 5. Результат работы программы из индивидуального задания 2

### Контрольные вопросы

1. Что является модулем языка Python?

Файл, содержащий Python-код и определения, который может быть использован в других программах Python. Модули позволяют организовать код в более крупные и структурированные программы.

2. Какие существуют способы подключения модулей в языке Python?

Существует несколько способов подключения модулей в Python:

1) Использование ключевого слова `import` для подключения всего модуля.

2) Использование ключевого слова `from` для импорта конкретных объектов из модуля.

3) Использование ключевого слова `as` для создания псевдонимов при импорте модулей.

3. Что является пакетом языка Python?

Папка, которая содержит модули. Пакеты позволяют организовать модули в иерархическую структуру.

4. Каково назначение файла `__init__.py` ?

Файл `__init__.py` в пакете Python используется для указания, что каталог, в котором он находится, должен рассматриваться как пакет Python.

5. Каково назначение переменной `__all__` файла `__init__.py` ?

Переменная `_all_` в файле `_init_.py` используется для определения списка модулей, которые будут импортированы при использовании выражения `from package import`

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с модулями и пакетами языка программирования Python версии 3.x.