

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Северо-Кавказский федеральный университет»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2.13**

**по дисциплине «Основы программной инженерии»**

**Выполнил студент группы ПИЖ-б-о-20-1**

**Примаков В. Д. « »\_\_\_\_\_20\_\_г.**

**Подпись студента \_\_\_\_\_**

**Работа защищена « »\_\_\_\_\_20\_\_г.**

**Проверил Воронкин Р.А. \_\_\_\_\_**

**(подпись)**

## ВЫПОЛНЕНИЕ

### Задание 1

#### 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`. Номер варианта уточнить у преподавателя.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from mod1 import func1

if __name__ == "__main__":
    a = int(input("a? "))
    b = int(input("b? "))
    test = func1(a, b)
    print(test())
```

```
def func1(a, b):
    x = a
    y = b

    def func2():
        nonlocal x, y
        return f"Для значений {x}, {y} функция f({x},{y}) = {x + y}"
    return func2
```

```
C:\Users\student-09-525\Desktop\Laba1\
a? 1
b? 4
Для значений 1, 4 функция f(1,4) = 5

Process finished with exit code 0
```

## Задание 2

### 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. Номер варианта уточнить у преподавателя.

```
ind2.py x ind1.py x mod1.py x
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   from bdata import *
5
6   if __name__ == '__main__':
7       menu.main()
8
```

```
ind2.py x __init__.py x
__all__ = ['get_goods', 'select_goods', 'menu', 'display_goods']
```

```
def display_goods(goods):
    """
    Отобразить список товаров.
    """
    print(goods)
    # Проверить, что список товаров не пуст.
    if goods:
        # Заголовок таблицы.
        line = '+--+--+--+--+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '|{:>4}|{:<30}|{:<20}|{:>8}|'.format(
                "№",
                "Название",
                "Магазин",
                "Цена"
            )
        )
        print(line)
        # Вывести данные о всех товарах.
        for idx, good in enumerate(goods, 1):
            print(
                '|{:>4}|{:<30}|{:<20}|{:>8}|'.format(
                    idx,
                    good.get('name', ''),
                    good.get('shop', ''),
                    good.get('price', 0)
                )
            )
            print(line)
    else:
        print("Список товаров пуст.")
```

```
1 import sys
2 from bdata.get_goods import get_goods
3 from bdata.display_goods import display_goods
4 from bdata.select_goods import select_goods
5
6 def main():
7     # Список товаров.
8     goods = []
9
10    # Организовать бесконечный цикл запроса команд.
11    while True:
12        # Запросить команду из терминала.
13        command = input(">>> ").lower()
14
15        # Выполнить действие в соответствие с командой.
16        if command == 'exit':
17            break
18
19        elif command == 'add':
20            # Запросить данные о товаре.
21            good = get_goods()
22
23            # Добавить словарь в список.
24            goods.append(good)
25            # Отсортировать список в случае необходимости.
26            if len(goods) > 1:
27                goods.sort(key=lambda item: item.get('name', ''))
28
29        elif command == 'list':
30            # Отобразить все товары.
31            display_goods(goods)
32
33        elif command.startswith('select '):
34            # Разбить команду на части для выделения стажа.
35            parts = command.split(' ', maxsplit=1)
36            # Получить требуемые товары.
37            shop = parts[1]
38
39            # Выбрать товары магазина.
40            selected = select_goods(goods, shop)
41            # Отобразить выбранные товары.
42            display_goods(selected)
43
44        elif command == 'help':
45            # Вывести справку о работе с программой.
46            print("Список команд:\n")
47            print("add - добавить товар;")
48            print("list - вывести список товаров;")
49            print("select <имя магазина> - запросить товары магазина;")
50            print("help - отобразить справку;")
51            print("exit - завершить работу с программой.")
52        else:
53            print(f"Неизвестная команда {command}", file=sys.stderr)
```

```
def get_goods():
    """
    Запросить данные о товаре.
    """
    name = input("Название товара: ")
    shop = input("Название магазина: ")
    price = float(input("Стоимость: "))
    # Создать словарь.
    return {
        'name': name,
        'shop': shop,
        'price': price,
    }
```

```

def select_goods(goods, shop):
    """
    Выбрать товары магазина.
    """

    # Счетчик записей.
    count = 0

    # Сформировать список товаров.
    result = []

    for good in goods:
        if shop == good.get('shop', shop):
            count += 1
            result.append(good)

    # Проверка на отсутствие товаров или выбранного магазина.
    if count == 0:
        print("Такого магазина не существует либо нет товаров.")
    else:
        # Возвратить список выбранных товаров.
        return result

```

Ссылки на репозитории

GitHub - [https://github.com/surai5a/laba\\_2\\_13](https://github.com/surai5a/laba_2_13)

Ответы на контрольные вопросы

1. Под модулем в Python понимается файл с расширением .py. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы: программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы.
2. Модули в Python можно импортировать конструкцией “**import** module\_name1, module\_name2”.
3. Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл \_\_init\_\_.py . Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).
4. Файл \_\_init\_\_.py вызывается при импорте пакета или модуля в пакете при условии его наличия в самом пакете.
5. Переменная \_\_all\_\_ содержит все модули пакета которые импортируются через конструкцию “**from** package\_name **import** \*”