

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

**Отчет по лабораторной работе № 1
«Модули и пакеты»**

по дисциплине «Основы программной инженерии»

Выполнила:
Первых Дарья Александровна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ

```
def hello_world():  
    print('Hello world!')
```

Рисунок 1 – Пример импорта и использования модуля math, который содержит математические функции

```
>>> import math, datetime  
>>> math.cos(math.pi/4)  
0.7071067811865476  
>>> datetime.date(2017, 3, 21)  
datetime.date(2017, 3, 21)
```

Рисунок 2 – Пример использования нескольких модулей

```
>>> import math as m  
>>> m.sin(m.pi/3)  
0.8660254037844386
```

Рисунок 3 – Пример задания псевдонима

```
>>> from math import cos  
>>> cos(3.14)  
-0.9999987317275395
```

Рисунок 4 – Пример импортирования отдельной функции

```
>>> from math import cos  
>>> cos(3.14)  
-0.9999987317275395  
>>> sin(3.14)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'sin' is not defined. Did you mean: 'bin'?  
>>> math.sin(3.14)  
0.0015926529164868282
```

Рисунок 5 – Пример импорта конкретного объекта

```
>>> from math import cos, sin, pi  
>>> cos(pi/3)  
0.5000000000000001  
>>> sin(pi/3)  
0.8660254037844386
```

Рисунок 6 – Пример импортирования нескольких функций из модуля

```
>>> from math import factorial as f
>>> f(4)
24
```

Рисунок 7 – Пример задания псевдонима импортируемому объекту

```
>>> from math import *
>>> cos(pi/2)
6.123233995736766e-17
>>> sin(pi/4)
0.7071067811865476
>>> factorial(6)
720
```

Рисунок 8 – Пример импортирования всех функций и классов

```
import fincalc.simper
fv = fincalc.simper.fv(pv, i, n)
import fincalc.simper as sp
fv = sp.fv(pv, i, n)
from fincalc import simper
fv = simper.fv(pv, i, n)
```

Рисунок 9 – Пример использования пакетов

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

import fun

if __name__ == "__main__":
    figure_type = int(input("0 - прямоугольник, 1 - треугольник: "))
    a, b = input("Введите два параметра фигуры: ").split()
    print(fun.figure(figure_type)(int(a), int(b)))
```

Рисунок 10 – Основной файл задания 1

```

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

def figure(type=0):
    def result(a, b):
        if type:
            return (a * b) / 2
        else:
            return a * b
    return result

```

Рисунок 11 – Модуль задания 1

```

C:\Users\podar\study\anaconda\envs\LR16
0 - прямоугольник, 1 - треугольник: 1
Введите два параметра фигуры: 4 6
12.0

```

Рисунок 12 – Результат работы программы

```

from moduli import add, list, help, select

if __name__ == '__main__':
    products = []

    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break

        elif command == 'add':
            products.append(add.add())

        elif command == 'list':
            list.list(products)

```

Рисунок 13 – Основной файл задания 2

```

__all__ = ["__init__.py", "list", "select", "help"]

```

Рисунок 14 – Файл _init_.py задания 2

```
def add():  
    name = input("Название продукта: ")  
    shop = input("Название магазина: ")  
    price = float(input("Стоимость: "))  
    products = {  
        'name': name,  
        'shop': shop,  
        'price': price,  
    }  
    return products
```

Рисунок 15 – Модуль add задания 2

```
def help():  
    print("Список команд:\n")  
    print("add - добавить запись;")  
    print("list - вывести список;")  
    print("select <продукт> - запросить информацию о продукте;")  
    print("help - отобразить справку;")  
    print("exit - завершить работу с программой.")
```

Рисунок 16 – Модуль help задания 2

```

def list(products):
    if products:
        line = '--{}--{}--{}--{}--'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "№",
                "Продукт",
                "Магазин",
                "Цена"
            )
        )
        print(line)

    for idx, product in enumerate(products, 1):
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                idx

```

Рисунок 17 – Модуль list задания 2

```

def select_products(products):
    products = input("Введите товар: ")
    result = []
    for tovar in products:
        if tovar.get('name', ''):
            result.append(tovar)
    return result

```

Рисунок 18 – Модуль select задания 2

```
LR16 9 products = []
10
11 while True:
12     command = input(">>> ").lower()
13
14     if command == 'exit':
15         break
16
17 if __name__ == '__main__':
18     while True:
19         elif command.startswith('select...'):
```

Python Shell (zd2) ×

Стоимость: 70

>>> add

Название продукта: манго

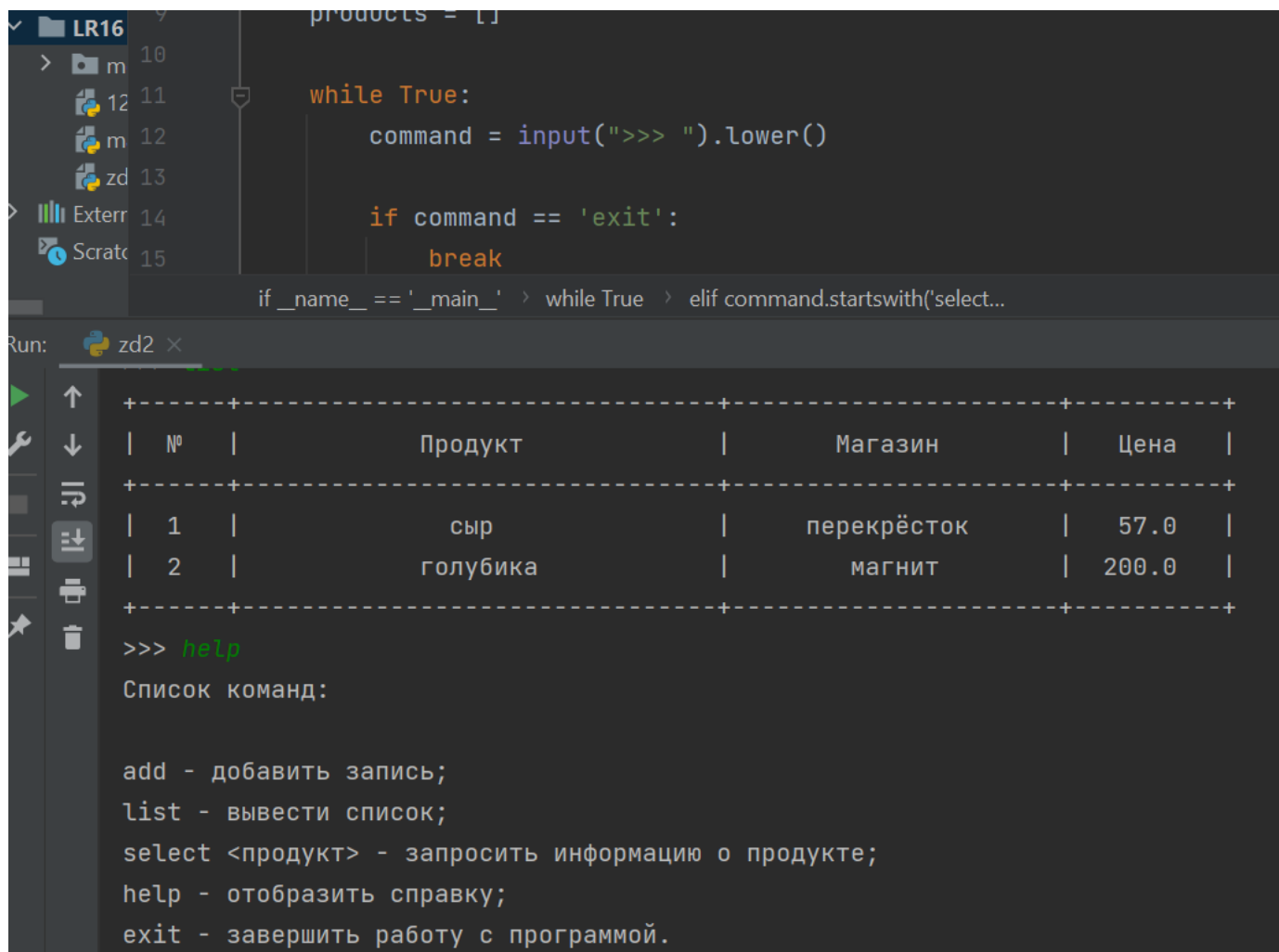
Название магазина: пятерочка

Стоимость: 150

>>> list

№	Продукт	Магазин	Цена
1	голубика	перекресток	200.0
2	киви	магнит	148.4
3	апельсин	перекресток	100.0
4	бананы	магнит	70.0
5	манго	пятерочка	150.0

Рисунок 19 – Пример работы программы



The image shows a Python IDE with a dark theme. The top pane displays a code editor with the following Python code:

```
products = []  
  
while True:  
    command = input(">>> ").lower()  
  
    if command == 'exit':  
        break
```

The bottom pane shows a terminal window titled "Run: zd2". It displays the output of the program, which is a table of products and their prices. The table has four columns: "№", "Продукт", "Магазин", and "Цена". The data rows are:

№	Продукт	Магазин	Цена
1	сыр	перекрёсток	57.0
2	голубика	магнит	200.0

Below the table, the terminal shows the user inputting `>>> help`, followed by the program's response: "Список команд:" (List of commands:). The list of commands is:

- add - добавить запись;
- list - вывести список;
- select <продукт> - запросить информацию о продукте;
- help - отобразить справку;
- exit - завершить работу с программой.

Рисунок 20 – Пример работы программы


```
products = []

while True:
    command = input(">>> ").lower()

    if command == 'exit':
        break

if __name__ == '__main__':
```

Terminal Output:

```
>>> add
Название продукта: сыр
Название магазина: перекрёсток
Стоимость: 57
>>> add
Название продукта: голубика
Название магазина: магнит
Стоимость: 200
>>> list
```

№	Продукт	Магазин	Цена
1	сыр	перекрёсток	57.0
2	голубика	магнит	200.0

Рисунок 21 – Пример работы программы

ВОПРОСЫ

1. Что является модулем языка Python?

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.

2. Какие существуют способы подключения модулей в языке Python?

```
import math
import math, datetime
import math as m from
math import cos

from math import cos, sin from
math import factorial as ffrom
math import *
```

3. Что является пакетом языка Python?

Пакет – это каталог, включающий в себя другие каталоги и модули, но при

этом дополнительно содержащий файл `__init__.py`. Пакеты используются для формирования пространства имён, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла `__init__.py`?

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию.

Пример: `from имя_пакета import *`

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

Она хранит список модулей, который импортируется при загрузке через конструкцию `from имя_пакета import *`. К примеру, данная переменная может выглядеть следующим образом: `__all__ = ["simper", "compper", "annuity"]`