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 №11

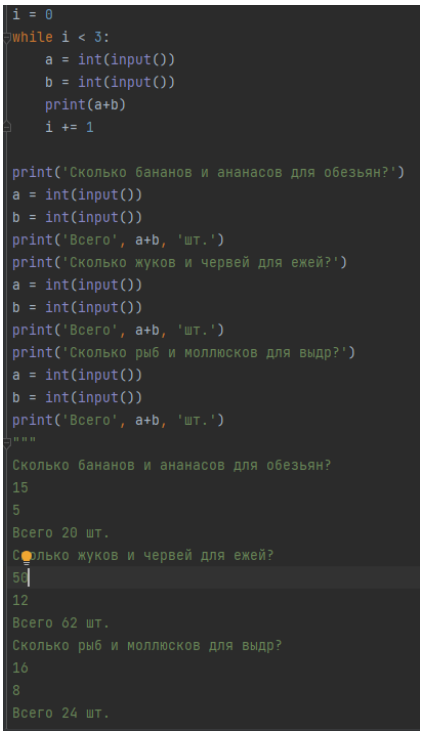
по дисциплине основы программной инженерии

Выполнил: Духно Михаил Александрович,

2 курс, группа ПИЖ-б-о-20-1,

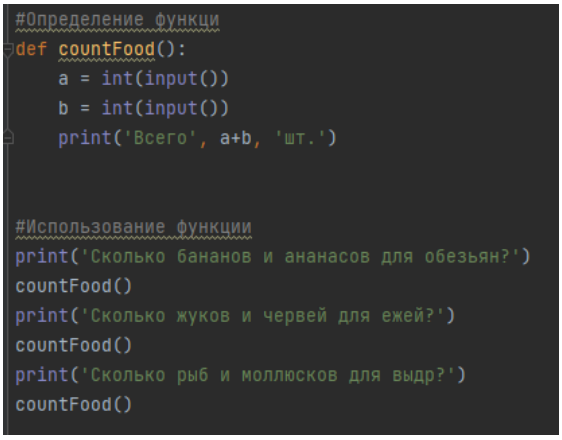
Проверил: Доцент кафедры инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2021 г



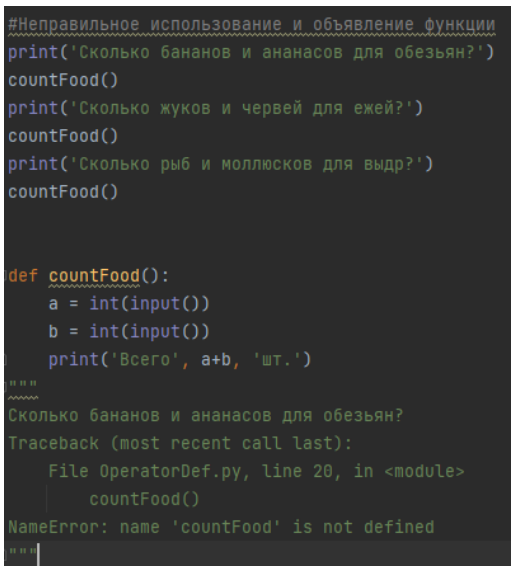
```
i = 0
while i < 3:
    a = int(input())
    b = int(input())
    print(a+b)
    i += 1

print('Сколько бананов и ананасов для обезьян?')
a = int(input())
b = int(input())
print('Всего', a+b, 'шт.')
print('Сколько жуков и червей для ежей?')
a = int(input())
b = int(input())
print('Всего', a+b, 'шт.')
print('Сколько рыб и моллюсков для выдр?')
a = int(input())
b = int(input())
print('Всего', a+b, 'шт.')
"""
Сколько бананов и ананасов для обезьян?
15
5
Всего 20 шт.
Сколько жуков и червей для ежей?
56
12
Всего 62 шт.
Сколько рыб и моллюсков для выдр?
16
8
Всего 24 шт.
```

Рисунок 11.1 – Функции в программировании



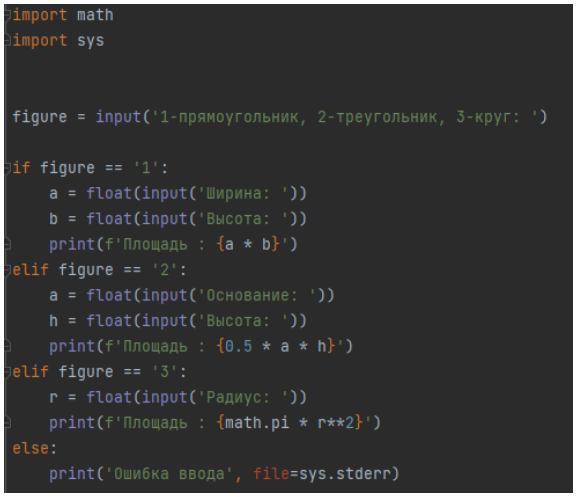
```
#Определение функци
def countFood():
    a = int(input())
    b = int(input())
    print('Всего', a+b, 'шт.')


#Использование функции
print('Сколько бананов и ананасов для обезьян?')
countFood()
print('Сколько жуков и червей для ежей?')
countFood()
print('Сколько рыб и моллюсков для выдр?')
countFood()
```

Рисунок 11.2 – Оператор def (1)



```
#Неправильное использование и объявление функции
print('Сколько бананов и ананасов для обезьян?')
countFood()
print('Сколько жуков и червей для ежей?')
countFood()
print('Сколько рыб и моллюсков для выдр?')
countFood()


def countFood():
    a = int(input())
    b = int(input())
    print('Всего', a+b, 'шт.')
"""
Сколько бананов и ананасов для обезьян?
Traceback (most recent call last):
    File OperatorDef.py, line 20, in <module>
        countFood()
NameError: name 'countFood' is not defined
"""
```

Рисунок 11.3 – Оператор def (2)



```
import math
import sys


figure = input('1-прямоугольник, 2-треугольник, 3-круг: ')

if figure == '1':
    a = float(input('Ширина: '))
    b = float(input('Высота: '))
    print(f'Площадь : {a * b}')
elif figure == '2':
    a = float(input('Основание: '))
    h = float(input('Высота: '))
    print(f'Площадь : {0.5 * a * h}')
elif figure == '3':
    r = float(input('Радиус: '))
    print(f'Площадь : {math.pi * r**2}')
else:
    print('Ошибка ввода', file=sys.stderr)
```

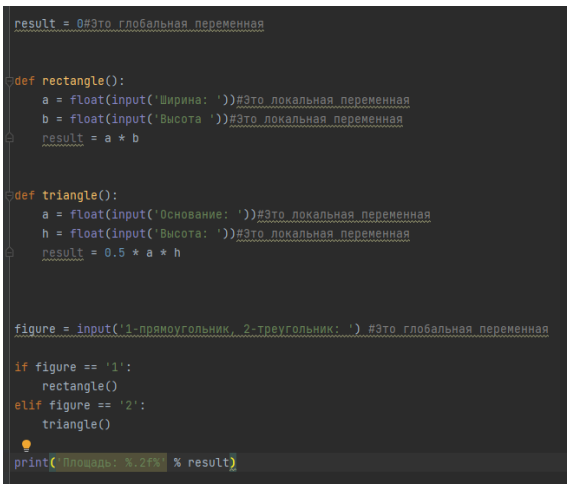
Рисунок 11.4 – Функции придают программе струтуру (1)

```python
import math
import sys


def rectangle():
    a = float(input('Ширина: '))
    b = float(input('Высота '))
    print(f' {a * b}')


def triangle():
    a = float(input('Основание: '))
    h = float(input('Высота: '))
    print(f' {0.5 * a * h}')


def circle():
    r = float(input('Радиус: '))
    print(f' {math.pi * r**2}')


figure = input('1-прямоугольник, 2-треугольник, 3-круг: ')

if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
elif figure == '3':
    circle()
else:
    print('Ошибка ввода', file=sys.stderr)
```

Рисунок 11.5 – Функции придают программе струтуру (2)

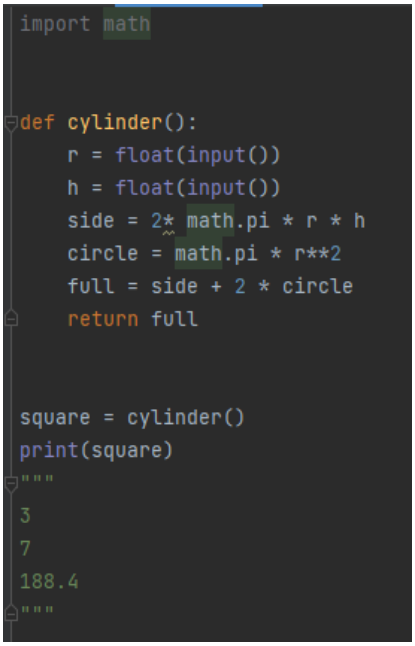
```python
def rectangle():
    a = float(input('Ширина: '))#Это локальная переменная
    b = float(input('Высота '))#Это локальная переменная
    print(f' {a * b}')


def triangle():
    a = float(input('Основание: '))#Это локальная переменная
    h = float(input('Высота: '))#Это локальная переменная
    print(f' {0.5 * a * h}')



figure = input('1-прямоугольник, 2-треугольник: ') #Это глобальная переменная

if figure == '1':
    rectangle()
elif figure == '2':
    triangle()
```

Рисунок 11.6 – Локальные и глобальные переменные (1)



```
result = 0#Это глобальная переменная


def rectangle():
    a = float(input('Ширина: '))#Это локальная переменная
    b = float(input('Высота '))#Это локальная переменная
    result = a * b


def triangle():
    a = float(input('Основание: '))#Это локальная переменная
    h = float(input('Высота: '))#Это локальная переменная
    result = 0.5 * a * h



figure = input('1-прямоугольник, 2-треугольник: ') #Это глобальная переменная

if figure == '1':
    rectangle()
elif figure == '2':
    triangle()

print('Площадь: %.2f%' % result)
```

Рисунок 11.7 – Локальные и глобальные переменные (2)



```
import math


def cylinder():
    r = float(input())
    h = float(input())
    side = 2* math.pi * r * h
    circle = math.pi * r**2
    full = side + 2 * circle
    return full


square = cylinder()
print(square)
"""
3
7
188.4
"""
```

Рисунок 11.8 – Оператор return (1)

```python
import math


def cylinder():
    try:
        r = float(input())
        h = float(input())
    except ValueError:
        return

    side = 2* math.pi * r * h
    circle = math.pi * r**2
    full = side + 2 * circle
    return full


print(cylinder())
```

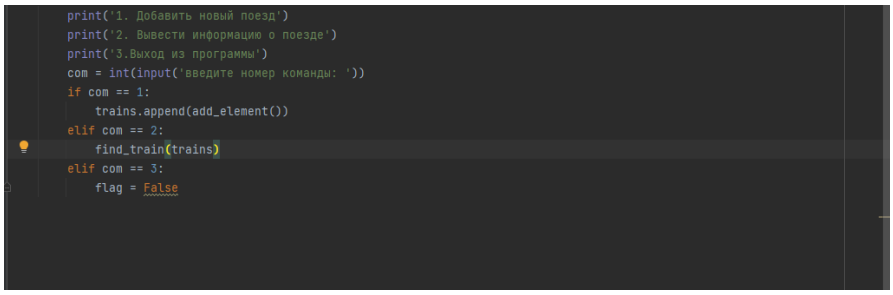
Рисунок 11.9 – Оператор return (2)

```python
#Возврат нескольких значений
import math


def cylinder():
    r = float(input())
    h = float(input())
    side = 2 * math.pi * r * h
    circle = math.pi * r**2
    full = side + 2 * circle
    return side, full


scyl, fcyl = cylinder()
print(f'Площадь боковой поверхности{scyl}')
print(f'Полная площадь{fcyl}')
```

Рисунок 11.10 – Оператор return (3)

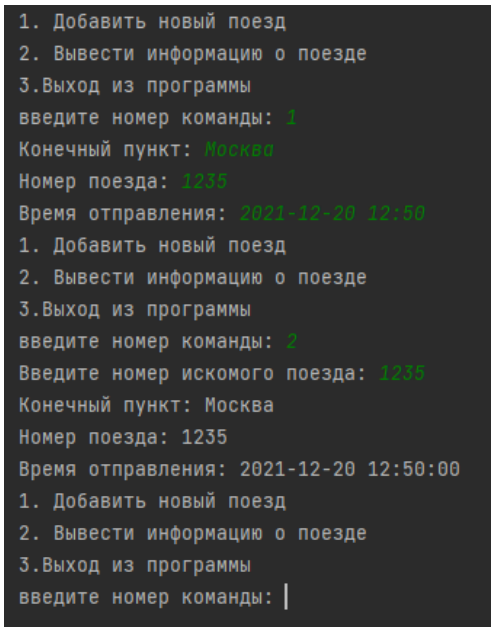
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from datetime import datetime


def add_element():
    name = input('Конечный пункт: ')
    num = input('Номер поезда: ')
    tm = datetime.strptime(input('Время отправления: '), '%Y-%m-%d %H:%M')
    trains = {}
    trains['name'] = name
    trains['num'] = int(num)
    trains['tm'] = tm
    return trains


def find_train(trains):
    num = int(input('Введите номер искомого поезда: '))
    for dcts in trains:
        if dcts['num'] == num:
            print(f'Конечный пункт: {dcts["name"]} \n'
                  f'Номер поезда: {dcts["num"]} \n'
                  f'Время отправления: {(dcts["tm"])}')
            return
    print('Поезда с таким номером нет')


if __name__ == '__main__':
    flag = True
    trains = []
    while flag:
```

Рисунок 11.11 – Код программы индивидуального задания (1)



```python
        print('1. Добавить новый поезд')
        print('2. Вывести информацию о поезде')
        print('3.Выход из программы')
        com = int(input('введите номер команды: '))
        if com == 1:
            trains.append(add_element())
        elif com == 2:
            find_train(trains)
        elif com == 3:
            flag = False
```

Рисунок 11.12 – Код программы индивидуального задания (2)



```
1. Добавить новый поезд
2. Вывести информацию о поезде
3.Выход из программы
введите номер команды: 1
Конечный пункт: Москва
Номер поезда: 1235
Время отправления: 2021-12-20 12:50
1. Добавить новый поезд
2. Вывести информацию о поезде
3.Выход из программы
введите номер команды: 2
Введите номер искомого поезда: 1235
Конечный пункт: Москва
Номер поезда: 1235
Время отправления: 2021-12-20 12:50:00
1. Добавить новый поезд
2. Вывести информацию о поезде
3.Выход из программы
введите номер команды:
```

Рисунок 11.13 – Результат работы программы

Контрольные вопросы:

1. Каково назначение функций в языке программирования Python?

Они нужны для упрощения кода и структурирования программы

2. Каково назначение операторов def и return?

Def – создать функцию

Return – вернуть значение из функции или выйти из функции до её конца

3. Каково назначение локальных и глобальных переменных при написании функций в Python?

Глобальные – их можно вызвать откуда угодно, они нужны чтобы хранить значение, которое используется в двух и более функциях

Локальные нужны чтобы хранить значение, актуальное исключительно для текущей функции

4. Как вернуть несколько значений из функции python?

return a, b

5. Какие существуют способы передачи значений в функцию?

При помощи объявления параметров при создании функции и дальнейшей передаче аргументов во время вызова

6. Как задать значение аргументов функции по умолчанию?

При создании функции присвоить нужное значение параметрам

def funct(par1='this is default meaning')

7.Каково назначение lambda-выражений в языке python?

Они нужны для написания коротких и простых функций, которые сразу же вызываются

8. Как осуществляется документирование кода согласно PEP257?

Оно осуществляется путём внесения комментария в тройные двойные кавычки

9. В чем особенность однострочных и многострочных форм строк документации?

Однострочные используются для краткого описания переменной или условных операторов. Многострочные используются для документирования функций и многострочных комментариев по поводу работы программы в целом