

**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 №11 по
дисциплине: основы программной инженерии**

Выполнил:

студент группы ПИЖ-б-о-21-1

Прокопов Дмитрий Владиславович

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2021 г.

Пример

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5      from datetime import date
6
7
8      def get_worker():
9          """
10         Запросить данные о работнике.
11         """
12
13         name = input("Фамилия и инициалы? ")
14         post = input("Должность? ")
15         year = int(input("Год поступления? "))
16
17         # Создать словарь.
18         return {
19             'name': name,
20             'post': post,
21             'year': year,
22         }
23
24
25     def display_workers(staff):
26         """
27         Отобразить список работников.
28         """
29
30         # Проверить, что список работников не пуст.
31         if staff:
32             # Заголовок таблицы.
33             line = '+-{}-+-{}-+-{}-+-{}-+'.format(
34                 '-' * 4,
35                 '-' * 30,
36                 '-' * 20,
37                 '-' * 8
38             )
39
40             print(line)
```

```

41         print(
42             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
43                 "№",
44                 "Ф.И.О.",
45                 "Должность",
46                 "Год"
47             )
48         )
49         print(line)
50
51         # Вывести данные о всех сотрудниках.
52         for idx, worker in enumerate(staff, 1):
53             print(
54                 '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
55                     idx,
56                     worker.get('name', ''),
57                     worker.get('post', ''),
58                     worker.get('year', 0)
59                 )
60             )
61             print(line)
62
63         else:
64             print("Список работников пуст.")
65
66
67     def select_workers(staff, period):
68         """
69         Выбрать работников с заданным стажем.
70         """
71
72         # Получить текущую дату.
73         today = date.today()
74
75         # Сформировать список работников.
76         result = []
77         for employee in staff:
78             if today.year - employee.get('year', today.year) >= period:
79                 result.append(employee)

```

```
80
81     # Возвратить список выбранных работников.
82     return result
83
84
85 def main():
86     """
87     Главная функция программы.
88     """
89
90     # Список работников.
91     workers = []
92
93     # Организовать бесконечный цикл запроса команд.
94     while True:
95         # Запросить команду из терминала.
96         command = input(">>> ").lower()
97
98         # Выполнить действие в соответствие с командой.
99         if command == 'exit':
100             break
101
102         elif command == 'add':
103             # Запросить данные о работнике.
104             worker = get_worker()
105
106             # Добавить словарь в список.
107             workers.append(worker)
108             # Отсортировать список в случае необходимости.
109             if len(workers) > 1:
110                 workers.sort(key=lambda item: item.get('name', ''))
111
112         elif command == 'list':
113             # Отобразить всех работников.
114             display_workers(workers)
115
116         elif command.startswith('select '):
117             # Разбить команду на части для выделения стажа.
118             parts = command.split(' ', maxsplit=1)
```

```
parts = command.split(' ', maxsplit=1)
# Получить требуемый стаж.
period = int(parts[1])

# Выбрать работников с заданным стажем.
selected = select_workers(workers, period)
# Отобразить выбранных работников.
display_workers(selected)

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <стаж> - запросить работников со стажем;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

if __name__ == '__main__':
    >>> add
    Фамилия и инициалы? Иванович С.У.
    Должность? ученый
    Год поступления? 2000

    >>> list
```

№	Ф.И.О.	Должность	Год
1	Иванович С.У.	ученый	2000
2	Сорокин И.В.	охранник	1992

Задание 8

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

def positive():
    print("Positive")

def negative():
    print("Negative")

def test():
    a = int(input("Put int num: "))
    if a > 0:
        positive()
    else:
        negative()

# def positive():
#     print("Positive")
#
#
# def negative():
#     print("Negative")

▶ if __name__ == '__main__':
    test()
```

C:\Users\dmidt\PycharmProjects\lb11
Put int num: 4
Positive

Задание 10

```

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

PI = 3.14

def circle(r):
    return PI * (r * r)

def cylinder():
    h = float(input("Put high of cylinder: "))
    r = float(input("Put radius of cylinder: "))
    x = int(input("1 - Calc only side area\n2 - Calc full area\n"))

    if x == 1:
        area = 2 * PI * r * h
    elif x == 2:
        area = 2 * PI * r * h
        area = 2 * (circle(r))
    print(area)

if __name__ == '__main__':
    cylinder()

```

Put high of cylinder: 4
 Put radius of cylinder: 5
 1 - Calc only side area
 2 - Calc full area
 2
 157.0

Put high of cylinder: 4
 Put radius of cylinder: 5
 1 - Calc only side area
 2 - Calc full area
 1
 125.60000000000001

Задание 12

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

def mult():
    x = 1
    while True:
        a = int(input("Put numbers you want mult(0 - stop): "))
        if a == 0:
            break
        else:
            x = x * a
    if x == 1:
        exit()
    else:
        print(x)

if __name__ == '__main__':
    mult()
```

C:\Users\dmidt\PycharmProjects\lb11\venv\Scripts>

Put numbers you want mult(0 - stop): 4

Put numbers you want mult(0 - stop): 5

Put numbers you want mult(0 - stop): 4

Put numbers you want mult(0 - stop): 3

Put numbers you want mult(0 - stop): 7

Put numbers you want mult(0 - stop): 0

1680

Задание 14


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

def get_input():
    a = input("Put some string: ")
    return a

def test_input(a):
    return a.isnumeric()

def str_to_int(a):
    return int(a)

def print_int(a):
    print(a)

if __name__ == '__main__':
    x = get_input()
    if test_input(x) == 1:
        x = str_to_int(x)
        print_int(x)
```

C:\Users\dmidt\PycharmProjects\lb11
Put some string: 48261
48261

Индивидуальное задание

```

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

schedule = []

def data_input():
    while True:
        global schedule
        schedule.append({
            'Ф.И.О.': input('Ф.И.О.? - '),
            'номер группы': int(input('Номер группы? - ')),
            'успеваемость': input('Успеваемость? - ')
        })
        if input('Напишите "д" чтобы продолжить ввод данных, "н" для завершения ввода.\n') == 'д':
            pass
        else:
            break
    schedule = sorted(schedule, key=lambda row: row['номер группы'])
    for i in schedule:
        print(i)

def data_search():
    destination = input('Студент который вас интересует?\n')
    approved = []
    for i in schedule:
        if i['Ф.И.О.'] == destination:
            approved.append(i)
    for i in approved:
        print(i)
    if not approved:
        print("По вашему запросу ничего не найдено.\n")

if __name__ == "__main__":
    x = int()
    while True:
        print("1 - Ввести данные\n"
              "2 - Поиск\n"
              "3 - Выход")
        x = int(input())
        if x == 1:
            data_input()
        elif x == 2:
            data_search()
        elif x == 3:
            break

```

```

C:\Users\dmidt\PycharmProjects\lb11\venv\Scripts\python.exe C:/Users/dmidt/PycharmP
1 - Ввести данные
2 - Поиск
3 - Выход
1
Ф.И.О.? - Аверин А.С.
Номер группы? - 2
Успеваемость? - 5
Напишите "д" чтобы продолжить ввод данных, "н" для завершения ввода.
д
Ф.И.О.? - Иванович. К.Ф.
Номер группы? - 4
Успеваемость? - 4
Напишите "д" чтобы продолжить ввод данных, "н" для завершения ввода.
н
{'Ф.И.О.': 'Аверин А.С.', 'номер группы': 2, 'успеваемость': '5'}
{'Ф.И.О.': 'Иванович. К.Ф.', 'номер группы': 4, 'успеваемость': '4'}
1 - Ввести данные
2 - Поиск
3 - Выход
2
Студент который вас интересует?
Аверин А.С.
{'Ф.И.О.': 'Аверин А.С.', 'номер группы': 2, 'успеваемость': '5'}
1 - Ввести данные
2 - Поиск
3 - Выход
3
Process finished with exit code 0

```

Контрольные вопросы

1. Функции решают проблему дублирования кода в разных местах программы. Благодаря им, есть возможность один и тот же участок кода не сразу, а когда понадобится.
2. Функции определяются оператором def. Return возвращает значение, вычисленное функцией в основное тело программы для его дальнейшей обработки.
3. Локальные и глобальные переменные призваны разграничить доступ к переменным между частями кода. Так глобальные переменные доступны в любом месте кода, когда локальные могут быть использованы, например, только в функции, если они были объявлены внутри неё.
4. Вернуть несколько значений сразу можно перечислив их через запятую после оператора return.

5. Передать значение в функцию можно передав имя переменных в качестве параметра, либо передав сами значения переменных.
6. Чтобы задать значение аргументов функции по умолчанию следует после объявлений всех параметров указать те, которые принимают значения по умолчанию в случае, если их значения не будут указаны (`def funct(a, b, c = 2)`)).
7. Лямбды – те же функции, но с упрощенным синтаксисом, и по сути являются выражениями. Они могут быть использованы там, где не могут функции, внутри литералов или в вызовах функций.
8. Документирование кода по PEP257 предусматривает использование тройных двойных кавычек. Также существует две формы строк документации: однострочная и многострочная.
9. Однострочная строка документации не должна быть "подписью" параметров функции / метода. Этот тип строк документации подходит только для C функций, где интроспекция не представляется возможной. Многострочные строки документации состоят из однострочной строки документации с последующей пустой строкой, а затем более подробным описанием. Первая строка может быть использована автоматическими средствами индексации, поэтому важно, чтобы она находилась на одной строке и была отделена от остальной документации пустой строкой.