

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2.8**

Работа с множествами в языке Python

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2022

## Работа со словарями в языке Python.

**Цель работы:** приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.10.

### Порядок выполнения работы:

#### 1. Проработка примеров:

Код программы:

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

import sys
from datetime import date

def get_worker():
    """
    Запросить данные о работнике.
    """
    name = input("Фамилия и инициалы: ")
    post = input("Должность: ")
    year = int(input("Год поступления: "))

    # Создать словарь
    return {
        'name': name,
        'post': post,
        'year': year,
    }

def display_worker(staff):
    """
    Отобразить список работников
    """
    # Проверить что список работников не пуст
    if staff:
        # Заголовок таблицы
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "№",
                "Ф.И.О.",
                "Должность",
                "Год"
            )
        )

    # Вывести данные о всех сотрудниках
    for idx, worker in enumerate(staff, 1):
```

```

        print(
            '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                idx,
                worker.get('name', ''),
                worker.get('post', ''),
                worker.get('year', 0)
            )
        )
        print(line)

    else:
        print("Список работников пуст.")

def select_workers(staff, period):
    """
    Выбрать работников с заданным стажем
    """
    # Получить текущую дату
    today = date.today()
    # Сформировать список работников
    result = []
    for employee in staff:
        if today.year - employee.get('year', today.year) >= period:
            result.append(employee)

    # Возвратить список выбранных работников
    return result

def main():
    """
    Главная функция программы
    """
    # Список работников
    workers = []

    # Организовать бесконечный цикл запроса команд
    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break

        elif command == 'add':
            worker = get_worker()
            workers.append(worker)

            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))

        elif command == 'list':
            display_worker(workers)

        elif command.startswith('select '):
            parts = command.split(' ', maxsplit=1)
            period = int(parts[1])
            selected = select_workers(workers, period)
            display_worker(selected)

        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить работника;")
            print("list - вывести список работников;")

```

```

        print("select <стаж> - запросить работников со стажем;")
        print("help - вывести список команд;")
        print("exit - завершить работу с программой.")

    else:
        print(f"Неизвестная команда {command}", file=sys.stderr)

if __name__ == '__main__':
    main()

```

Результат работы программы:

```

"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\python.exe"
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - вывести список команд;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы: Мальцев Н.А.
Должность: Студент
Год поступления: 2003
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |   Год   |
|  1 | Мальцев Н.А.               | Студент              |  2003   |
+-----+-----+-----+-----+
>>> select 19
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |   Год   |
|  1 | Мальцев Н.А.               | Студент              |  2003   |
+-----+-----+-----+-----+
>>> exit

Process finished with exit code 0

```

Рисунок 1. Результат работы программы из примера 1

## 2. Выполнение задачи 1:

Код программы:

```

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

def test():
    """
    Основная функция, в которой запрашивается число
    """

```

```

"""
# Ввод числа
chis = int(input("Введите число: "))

# Вызов функции в зависимости от знака числа
if chis < 0:
    negative()
elif chis >= 0:
    positive()

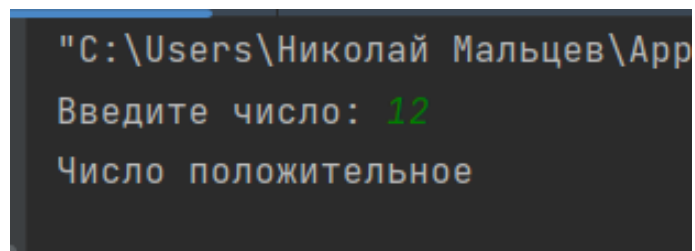
def negative():
    """
    Функция, вызываемая. если число отрицательное
    """
    print("Число отрицательное")

def positive():
    """
    Функция, вызываемая. если число положительное
    """
    print("Число положительное")

if __name__ == '__main__':
    test()

```

Результат работы программы:



```

C:\Users\Николай Мальцев\AppData
Введите число: 12
Число положительное

```

Рисунок 2. Результат работы программы к заданию 1

### 3. Выполнение задачи 2:

Листинг программы:

```

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

import math
import sys

def cylinder():
    """
    Вычисление площади полной или боковой поверхности цилиндра
    """
    while True:
        vs = int(input("Какую площадь вы хотите вычислит (введите число):\n"
                       "1 -> Боковой поверхности\n"
                       "2 -> Полной поверхности\n"
                       ">>> "))
        if (vs != 1) & (vs != 2):
            print(f"Неизвестная команда {vs}", file=sys.stderr)

```

```

        break

    r = int(input("Введите радиус "))
    h = int(input("Введите высоту цилиндра: "))

    if vs == 1:
        s = 2 * math.pi * r * h
        print("S(бок.) = ", '{:.3f}'.format(s))
        break

    elif vs == 2:
        s = (2 * math.pi * r * h) + (circle(r) * 2)
        print("S(полн.) = ", '{:.3f}'.format(s))
        break

def circle(r):
    """
    Вычисление площади круга по заданному радиусу
    """
    s = pow((math.pi * r), 2)
    return s

if __name__ == '__main__':
    cylinder()

```

Результат работы программы:

```

Какую площадь вы хотите вычислит (введите число):
1 -> Боковой поверхности
2 -> Полной поверхности
>>> 2
Введите радиус 12
Введите высоту цилиндра: 4
S(полн.) = 3144.039

```

Рисунок 3. Результат работы программы к заданию 2

#### 4. Выполнение задачи 3:

Листинг программы:

```

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

def factor():
    """
    Умножение введённых чисел. пока не будет введён 0
    """
    print("Вводите числа: ")
    b = 1
    while True:
        a = int(input(">>> "))
        if a == 0:
            break

```

```

        else:
            b *= a
        print("Результат: ", b)

if __name__ == '__main__':
    factor()

```

Результат выполнения программы:

```

Вводите числа:
>>> 1
>>> 2
>>> 3
>>> 4
>>> 5
>>> 0
Результат: 120

```

Рисунок 4. Результат выполнения программы к задаче 3

## 5. Выполнение задачи 4:

Листинг программы:

```

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

import sys

def get_input():
    """
    Запрашивает ввод с клавиатуры и
    возвращает в основную программу полученную строку.
    """
    get_str = input("Введите число: ")
    return get_str

def test_input(b):
    """
    Проверяет, можно ли переданное ей значение
    преобразовать к числу. Если да - возвращает True.
    """
    if type(b) == int or type(b) == float:
        return True
    elif b.isnumeric():
        return True
    else:
        return False

def str_to_int(b):
    """
    Преобразует переданное значение к
    целочисленному типу.
    """
    c = int(b)

```

```

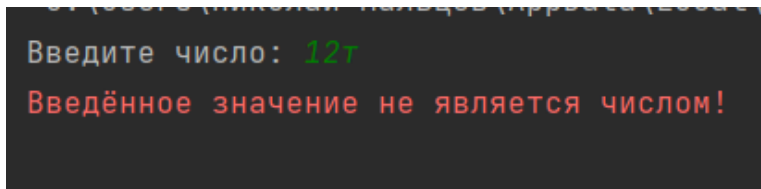
        return c

def print_int(c):
    """
    Выводит переданное значение на экран
    """
    print(c)

if __name__ == '__main__':
    a = get_input()
    # для теста print(type(a))
    bol = test_input(a)
    if bol:
        ch = str_to_int(a)
        # для теста print(type(ch))
        print_int(ch)
    else:
        print(f"Введённое значение не является числом!", file=sys.stderr)

```

Результат выполнения программы:



```

Введите число: 12r
Введённое значение не является числом!

```

Рисунок 5. Результат работы программы к задаче 4

## 6. Выполнение индивидуального задания (вариант 11):

Задача: решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.

Листинг программы:

```

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

import sys
import datetime

def add(list_man):
    # Запросить данные .
    name = input("Имя: ")
    number = input("Номер телефона ")
    date = input("Дата рождения: ")
    date = datetime.datetime.strptime(date, '%d.%m.%Y').date()

    # Создать словарь.
    man = {
        'name': name,
        'number': number,
        'date': date,
    }

    # Добавить словарь в список.
    list_man.append(man)

```



```

# Отсортировать список.
if len(list_man) > 1:
    list_man.sort(key=lambda item: item.get('date', ''))
return list_man

def list_d(list_man):
    # Заголовок таблицы.
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 20
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^20} |'.format(
            "No",
            "Имя",
            "Номер телефона",
            "Дата рождения"
        )
    )
    print(line)

    # Вывести данные о человеке.
    for idx, man in enumerate(list_man, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:<20} |'.format(
                idx,
                man.get('name', ''),
                man.get('number', ''),
                man.get('date', '')
            )
        )

    print(line)

def select(mans_list, sel_d):
    count = 0
    for man in mans_list:
        if man.get('number') == sel_d:
            count += 1
            print(
                '{:>4}: {}'.format(count, man.get('name', ''))
            )
            print('Номер телефона:', man.get('number', ''))
            print('Дата рождения:', man.get('date', ''))

    # Если счетчик равен 0, то человек не найден.
    if count == 0:
        print("Человек не найден.")

def help_d():
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить человека;")
    print("list - вывести список людей;")
    print("select <товар> - информация о человеке;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

```

```

if __name__ == '__main__':
    manlist = []
    while True:
        # Запросить команду
        command = input(">>> ").lower()
        if command == 'exit':
            break
        elif command == 'add':
            manlist = add(manlist)
        elif command == 'list':
            list_d(manlist)
        elif command.startswith('select '):
            parts = command.split(' ', maxsplit=1)
            sel = parts[1]
            select(manlist, sel)
        elif command == 'help':
            help_d()
        else:
            print("неизвестная команда {command}", file=sys.stderr)

```

Результат выполнения программы:

```

"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\python.exe
>>> help
Список команд:

add - добавить человека;
list - вывести список людей;
select <товар> - информация о человеке;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Имя: Nik
Номер телефона 198324
Дата рождения: 23.02.2003
>>> list
+-----+-----+-----+-----+
| No |          Имя          | Номер телефона | Дата р |
+-----+-----+-----+-----+
|  1 | Nik                   | 198324         | <20   |
+-----+-----+-----+-----+
>>> select 198324
1: Nik
Номер телефона: 198324
Дата рождения: 2003-02-23
>>> exit

Process finished with exit code 0

```

Рисунок 6. Результат выполнения программы к индивидуальному заданию

## **Ответы на вопросы:**

### **1. Каково назначение функций в языке программирования Python?**

Главной задачей функций в Python, как и в других языках программирования, является сокращение объёма кода и его структуризация. В функции, как правило, выносятся те части кода, которые выполняются в программе многократно.

### **2. Каково назначение операторов `def` и `return`?**

Оператор `def` необходим для определения функции. После него идёт название самой функции, передаваемые в функцию параметры и само тело функции. Оператор `return` служит для возвращения результата выполнения функции в основную программу, где эта функция была вызвана.

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

Локальные переменные существуют только внутри функции. В другой части программы как-либо вызывать или изменить их невозможно. Глобальные напротив – существуют во всей программе.

### **4. Как вернуть несколько значений из функции Python?**

После оператора `return` необходимо записать все возвращаемые переменные через запятую, а при вызове функции нужно задать необходимое количество переменных. Куда будут возвращены параметры.

### **5. Какие существуют способы передачи значений в функцию?**

По ссылке и по значению.

### **6. Как задать значение аргументов функции по умолчанию?**

Нужно в скобках передаваемых параметров присвоить им значение.

### **7. Каково назначение `lambda`-выражений в языке Python?**

`Lambda`-выражения – это небольшие функции, которые вызываются в программе один раз.

### **8. Как осуществляется документирование кода согласно PEP257?**

Если пояснение функции содержит одну строку, то достаточно двух кавычек с каждой стороны строки. Пример: `"""Пояснение"""`. Если это

многострочное пояснение, то необходимо три кавычки с каждой стороны. Пояснение находится в теле функции, сразу после её объявления.

**Вывод:** в ходе работы были изучены функции в языке Python3.10, получены навыки работы с ними.