

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8
дисциплины «Программирование на Python»

Выполнил:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики: кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с функциями в языке Python

Цель работы: приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Пример 1:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  from datetime import date
7
8
9  def get_worker():
10     """
11     Запросить данные о работнике.
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     # Проверить, что список работников не пуст.
30     if staff:
31         # Заголовок таблицы.
32         line = '+--{}-+-{}-+-{}-+-{}-+'.format(
33             '-' * 4,
34             '-' * 30,
35             '-' * 20,
36             '-' * 8
37         )
38         print(line)
39         print(
40             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
41                 "No",
42                 "Ф.И.О.",
43                 "Должность",
44                 "Год"
45             )
46         )
47         print(line)
```

```

49         # Вывести данные о всех сотрудниках.
50         for idx, worker in enumerate(staff, 1):
51             print(
52                 '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
53                     idx,
54                     worker.get('name', ''),
55                     worker.get('post', ''),
56                     worker.get('year', 0)
57                 )
58             )
59         print(line)
60
61     else:
62         print("Список работников пуст.")
63
64
65 def select_workers(staff, period):
66     """
67     Выбрать работников с заданным стажем.
68     """
69     # Получить текущую дату.
70     today = date.today()
71
72     # Сформировать список работников.
73     result = []
74     for employee in staff:
75         if today.year - employee.get('year', today.year) >= period:
76             result.append(employee)
77
78     # Возвратить список выбранных работников.
79     return result
80
81
82 def main():
83     """
84     Главная функция программы.
85     """
86     # Список работников.
87     workers = []
88
89     # Организовать бесконечный цикл запроса команд.
90     while True:
91         # Запросить команду из терминала.
92         command = input(">>> ").lower()
93
94         # Выполнить действие в соответствие с командой.
95         if command == 'exit':
96             break

```

```

97
98     elif command == 'add':
99         # Запросить данные о работнике.
100         worker = get_worker()
101
102         # Добавить словарь в список.
103         workers.append(worker)
104         # Отсортировать список в случае необходимости.
105         if len(workers) > 1:
106             workers.sort(key=lambda item: item.get('name', ''))
107
108     elif command == 'list':
109         # Отобразить всех работников.
110         display_workers(workers)
111
112     elif command.startswith('select '):
113         # Разбить команду на части для выделения стажа.
114         parts = command.split(' ', maxsplit=1)
115         # Получить требуемый стаж.
116         period = int(parts[1])
117
118         # Выбрать работников с заданным стажем.
119         selected = select_workers(workers, period)
120         # Отобразить выбранных работников.
121         display_workers(selected)
122
123     elif command == 'help':
124         # Вывести справку о работе с программой.
125         print("Список команд:\n")
126         print("add - добавить работника;")
127         print("list - вывести список работников;")
128         print("select <стаж> - запросить работников  стажем;")
129         print("help - отобразить справку;")
130         print("exit - завершить работу  программой.")
131
132     else:
133         print(f"Неизвестная команда {command}", file=sys.stderr)
134
135
136 if __name__ == '__main__':
137     main()

```

Индивидуальное задание:

Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.

```

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_workers(staff):
    """
    Отобразить список работников.
    """
    if staff:
        line = '+-{}-+-{}-+-{}-+-{}-+'.format('-' * 4, '-' * 30, '-' * 20, '-' * 8)
        print(line)
        print(' | {:^4} | {:^30} | {:^20} | {:^8} |'.format("No", "Ф.И.О.", "Должность", "Год"))
        print(line)

        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.
    """
    today = date.today()
    result = []

    for employee in staff:
        if today.year - employee.get('year', today.year) >= period:
            result.append(employee)

    return result
```

```

def main():
    """
    Главная функция программы.
    """
    workers = []

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

        if command == 'exit':
            break

        elif command == 'add':
            workers.append(get_worker())
            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))

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

        elif command.startswith('select '):
            parts = command.split(' ', maxsplit=1)
            period = int(parts[1])
            selected = select_workers(workers, period)
            display_workers(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 - отобразить справку;")
            print("exit - завершить работу с программой.")

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

if __name__ == '__main__':
    main()

```

Ответы на вопросы:

1. Каково назначение функций в языке программирования Python?
 - Функции в Python используются для группировки кода внутри блока, который можно вызывать многократно. Они облегчают повторное использование кода и делают программу более структурированной.

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

- Оператор ``def`` используется для определения функций, а оператор ``return`` - для возвращения значения из функции. ``def`` задает блок кода, а ``return`` указывает, что функция должна вернуть определенное значение.

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

- Локальные переменные существуют только внутри функции и не видны извне. Глобальные переменные объявляются вне функций и могут быть видны в любой части программы.

4. Как вернуть несколько значений из функции Python?

- Несколько значений можно вернуть, разделив их запятой: ``return value1, value2``. Эти значения будут упакованы в кортеж.

5. Какие существуют способы передачи значений в функцию?

- Значения могут передаваться в функцию позиционно, по имени (ключевым словом) или через аргументы со значениями по умолчанию.

6. Как задать значение аргументов функции по умолчанию?

- Значения аргументов по умолчанию задаются в определении функции, например: ``def my_function(arg1, arg2=default_value)``.

7. Каково назначение lambda-выражений в языке Python?

- Lambda-выражения создают анонимные функции. Они полезны для создания небольших функций, которые используются только в конкретном контексте.

8. Как осуществляется документирование кода согласно PEP257?

- Документация кода в Python обычно осуществляется с использованием строк документации (docstrings). PEP257 определяет стандарты для написания этих строк, включая форматирование и структуру.

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

- Однострочные строки документации обычно используются для краткого описания функции, класса или модуля. Многострочные строки документации обеспечивают более подробное описание и могут охватывать несколько строк кода.

Вывод В ходе работы приобрёл навыки по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.