

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 №2.8 по дисциплине основы программной
инженерии**

Выполнила:
Лобанова Елизавета Евгеньевна,
2 курс, группа ПИЖ-б-о-20-1

Проверил:
Доцент кафедры прикладной
математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2021 г.

Ход работы

```
#!/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_workers(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(
            "No",
            "Ф.И.О.",
            "Должность",
            "Год"
        )
    )
    print(line)
# Вывести данные о всех сотрудниках.
for idx, worker in enumerate(staff, 1):

```

```

    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:
        # Запросить команду из терминала.
        command = input(">>> ").lower()
        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break
        elif command == 'add':
            # Запросить данные о работнике.
            worker = get_worker()
            # Добавить словарь в список.
            workers.append(worker)
            # Отсортировать список в случае необходимости.
            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))
        elif command == 'list':
            # Отобразить всех работников.
            display_workers(workers)
        elif command.startswith('select '):
            # Разбить команду на части для выделения стажа.
            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__':
    main()

```

Пример 1

```

>>> add
Фамилия и инициалы? Вторых П.А.
Должность? Программист
Год поступления? 2019
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|   1 | Вторых П.А.              | Программист         |   2019  |
+-----+-----+-----+-----+

>>> select 3
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |   Год   |
+-----+-----+-----+-----+
|   1 | Вторых П.А.              | Программист         |   2019  |
+-----+-----+-----+-----+

>>> edit
>>> Неизвестная команда edit
exit

Process finished with exit code 0

```

Результат 1 программы (1)

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

def positive():
    print("Число положительное")

def negative():
    print("Число отрицательное")

def test():
    number = int(input("Введите целое число: "))
    if number > 0:
        positive()
    elif number < 0:
        negative()
    else:
        print("Число - 0")

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

Задание 1

Введите целое число: 345

Число положительное

Process finished with exit code 0

Результат 1 программы (1)

```
Введите целое число: -98
Число отрицательное

Process finished with exit code 0
```

Результат 1 программы (2)

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

from math import pi

def cylinder():

    def circle(rad):
        return pi * rad * rad

    r = int(input("Введите радиус: "))
    h = int(input("Введите высоту: "))
    choose = input("Площадь боковой поверхности цилиндра - a\n"
                  "Полная площадь цилиндра - b\n"
                  "a/b: ")

    if choose == 'a':
        print(f"Площадь боковой поверхности цилиндра = {2 * pi * r * h}")
    else:
        print(f"Полная площадь цилиндра = {2 * pi * r * h + 2 * circle(r)}")

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

Задание 2

```
Введите радиус: 33
Введите высоту: 12
Площадь боковой поверхности цилиндра - a
Полная площадь цилиндра - b
a/b: a
Площадь боковой поверхности цилиндра = 2488.1413816431163

Process finished with exit code 0
```

Результат 2 программы (1)

```
Введите радиус: 2
Введите высоту: 4
Площадь боковой поверхности цилиндра - a
Полная площадь цилиндра - b
a/b: b
Полная площадь цилиндра = 75.39822368615503

Process finished with exit code 0
```

Результат 2 программы (2)

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

def multi():
    number = int(input("Введите число: "))
    result = 1
    if number == 0:
        return None
    while number != 0:
        result *= number
        number = int(input("Введите число: "))
    return result

if __name__ == '__main__':
    print(f"Вызов функции и её результата = {multi()}")
```

Задание 3

```
Введите число: 44
Введите число: 76
Введите число: 48
Введите число: 2
Введите число: 0
Вызов функции и её результата = 321024

Process finished with exit code 0
```

Результат 3 программы (1)

```
Введите число: 8
Введите число: 9
Введите число: 0
Вызов функции и её результата = 72

Process finished with exit code 0
```

Результат 3 программы (2)


```

def get_input():
    return input()

def test_input(string):
    return string.isdigit()

def str_to_int(string):
    return int(string)

def print_int(integer):
    print(integer)

def main():
    data = get_input()
    if test_input(data):
        print_int(str_to_int(data))

if __name__ == '__main__':
    main()

```

Задание 4

```

kg

Process finished with exit code 0

```

Результат 4 программы (1)

```

38
38

Process finished with exit code 0

```

Результат 4 программы (2)

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

import sys
from datetime import datetime

def main():
    workers = []
    while True:
        command = get_command()
        if command == 'exit':
            break

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

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

        elif command.startswith('select'):
            select(command, workers)
```

```
        elif command == 'help':
            print_help()
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

def get_command():
    return input(">>> ").lower()

def add():
    name = input("Фамилия и имя: ")
    tel = input("Номер телефона: ")
    dateString = input("День рождения: ")
    worker = {
        'name': name,
        'tel': tel,
        'date': datetime.strptime(dateString, "%Y-%m-%d")
    }
    return worker

def print_list(workers):
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
```

```
        '-' * 4,  
        '-' * 30,  
        '-' * 20,  
        '-' * 16  
    )  
    print(line)  
    print(  
        '| {:^4} | {:^30} | {:^20} | {:^16} |'.format(  
            "No",  
            "Ф.И.",  
            "Телефон",  
            "Дата рождения"  
        )  
    )  
    print(line)  
    for idx, worker in enumerate(workers, 1):  
        print(  
            '| {:>4} | {:<30} | {:<20} | {:>16} |'.format(  
                idx,  
                worker.get('name', ''),  
                worker.get('tel', ''),  
                worker.get('date', 0)  
            )  
        )  
    )
```

```

    print(line)

def select(command, workers):
    parts = command.split(' ', maxsplit=1)
    period = int(parts[1])
    count = 0
    for worker in workers:
        if worker['date'].month == period:
            count += 1
            print(
                '{:>4}: {}'.format(count, worker.get('name', ''))
            )
    if count == 0:
        print("В этом месяце ни у одного из работников нет дня рождения.")

def print_help():
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <месяц> - месяц рождения работника;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

if __name__ == '__main__':
    main()

```

Индивидуальное задание 1

```

>>> add
Фамилия и имя: Васильев Петр
Номер телефона: 12-67-87
День рождения: 2000-12-23
>>> add
Фамилия и имя: Иванов Аркадий
Номер телефона: 33-67-42
День рождения: 1986-10-02
>>> select 11
В этом месяце ни у одного из работников нет дня рождения.
>>> select 12
      1: Васильев Петр
>>> list
+-----+-----+-----+
|  No  |          Ф.И.          |          Телефон          |
+-----+-----+-----+
|    1 | Васильев Петр          |    12-67-87              |
|    2 | Иванов Аркадий         |    33-67-42              |
+-----+-----+-----+

```

Результат индивидуального задания 1 (1)

Ответы на 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