

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

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5
дисциплины «**Программирование на Python**»
Вариант №11

Выполнила:

Ковжого Елизавета Андреевна

2 курс, группа ИВТ-б-о-24-1,

09.03.01 «Информатика и вычислительная техника», направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения

(подпись)

Проверил:

Воронкин Р. А., доцент
департамента цифровых,
робототехнических систем и
электроники института
перспективной инженерии.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: работа с множествами и словарями в языке Python.

Цель: приобретение навыков по работе с множествами и словарями при написании программ на языке программирования Python версии 3.x.

Порядок выполнения работы:

Адрес репозитория: https://github.com/LissKovzogo/Python_LAB_5.git

1. Создали, настроили и клонировали репозиторий Python_LAB_5.
2. Создали проект PyCharm в папке репозитория.
3. Проработали все примеры лабораторной работы и создали для каждого отдельный модуль Python.

Листинг кода pr_1.py:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    bn = u.difference(b)
    cn = u.intersection(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Листинг кода pr_2.py:

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

import sys

from datetime import date

if __name__ == '__main__':
    workers = []

    while True:

        command = input(">>> ").lower()

        if command == "exit":
            break

        elif command == "add":

            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))

            worker = {
                'name': name,
                'post': post,
                'year': year,
            }
            workers.append(worker)

        if len(workers) > 1:
```

```

        workers.sort(key=lambda item: item.get('name', ''))
elif command == 'list':
    line = '+-{}--{}--{}--{}+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 8
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
            "№",
            "ФИО",
            "Должность",
            "Год"
        )
    )
    print(line)

for idx, worker in enumerate(workers, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
            idx,
            worker.get('name', ''),
            worker.get('post', ''),
            worker.get('year', 0),
        )
    )
    print(line)

```

```
elif command.startswith('select '):
    today = date.today()
    parts = command.split(' ', maxsplit=1)
    period = int(parts[1])

    count = 0

    for worker in workers:
        if today.year - worker.get('year', today.year) >= period:
            count += 1
            print(
                '{:>4} : {}'.format(count, worker.get('name', ""))
            )
    if count == 0:
        print("Работники с заданным стажем не найдены.")

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)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    bn = u.difference(b)
    cn = u.intersection(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Рис. 1 — Работа с примером № 1

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import sys
from datetime import date
if __name__ == '__main__':
    workers = []

    while True:

        command = input(">>> ").lower()

        if command == "exit":
            break

        elif command == "add":

            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))

            worker = {
                'name': name,
                'post': post,
                'year': year,
            }
            workers.append(worker)

            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))
        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                *args: '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 8
            )
            print(line)
            print(
                '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(

```

Рис. 2 — Фрагмент примера № 2

```

print(line)

for idx, worker in enumerate(workers,1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
            *args: idx,
            worker.get('name', ''),
            worker.get('post', ''),
            worker.get('year', 0),
        )
    )
print(line)

elif command.startswith('select '):
    today = date.today()
    parts = command.split(sep: ' ', maxsplit=1)
    period = int(parts[1])

    count = 0

    for worker in workers:
        if today.year - worker.get('year', today.year) >= period:
            count += 1
            print(
                '| {:>4} : {}'.format(*args: count, worker.get('name', ''))
            )
    if count == 0:
        print("Работники с заданным стажем не найдены.")

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)

```

Рис. 3 — Фрагмент примера № 2

4. Выполнили индивидуальные задания согласно варианту.

Индивидуальное задание №1: Определить результат выполнения операций над множествами.

$$\begin{aligned}
A &= \{a, h, k\}; \\
B &= \{c, d, h, p, r\}; \\
C &= \{h, i, s\}; \\
D &= \{c, g, j, v, w\}; \\
X &= (A \cup B) \cap C; \\
Y &= (\bar{A} \cap \bar{B}) / (C \cup D).
\end{aligned}$$

Листинг кода задания №1:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"a", "h", "k"}
    b = {"c", "d", "h", "p", "r"}
    c = {"h", "i", "s"}
    d = {"c", "g", "j", "v", "w"}

    x = (a.union(b)).intersection(c)
    print(f"x = {x}")

    an = u.difference(a)
    bn = u.difference(b)

    y = (an.intersection(bn)).difference(c.union(d))
    print(f"y = {y}")
```

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      u = set("abcdefghijklmnopqrstuvwxyz")
5
6      a = {"a", "h", "k"}
7      b = {"c", "d", "h", "p", "r"}
8      c = {"h", "i", "s"}
9      d = {"c", "g", "j", "v", "w"}
10
11     x = (a.union(b)).intersection(c)
12     print(f"x = {x}")
13
14     an = u.difference(a)
15     bn = u.difference(b)
16
17     y = (an.intersection(bn)).difference(c.union(d))
18     print(f"y = {y}")
```

indiv_1 ×

C:\Users\Asus\AppData\Local\Microsoft\WindowsApps\python3.13.exe C:\Us
x = {'h'}
y = {'x', 'q', 'f', 't', 'e', 'm', 'n', 'z', 'l', 'b', 'o', 'u', 'y'}

Рис. 4 — Индивидуальное задание №1

Индивидуальное задание №2:

Использовать словарь, содержащий следующие ключи: фамилия, имя; номер телефона; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

Листинг кода задания №2:

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

```
import sys
from datetime import date

if __name__ == '__main__':
    contacts = []

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

        if command == "exit":
            break

        elif command == "add":
            name = input("Фамилия и инициалы? ")
            number = input("Номер телефона? ")
            birth_date = list(map(int, input("Дата рождения (в формате
XX.XX.XXXX)? ").split('.')))

            if number[0] == '+':
                number = '8' + number[2:]

            contact = {
                'name': name,
                'number': number,
                'date': birth_date,
            }
            contacts.append(contact)

        if len(contacts) > 1:
```

```
contacts.sort(key=lambda item: (item.get('date', [0, 0, 0])[2],
                                item.get('date', [0, 0, 0])[1],
                                item.get('date', [0, 0, 0])[0]))
```

```
elif command == 'list':
```

```
    line = '+-{}--{}--{}--{}+'.format(
```

```
        '-' * 4,
```

```
        '-' * 35,
```

```
        '-' * 20,
```

```
        '-' * 20
```

```
    )
```

```
    print(line)
```

```
    print(
```

```
        '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
```

```
            "№", "ФИО", "Номер телефона", "Дата рождения"
```

```
        )
```

```
    )
```

```
    print(line)
```

```
for idx, contact in enumerate(contacts, 1):
```

```
    date_list = contact.get('date', [0, 0, 0])
```

```
    date_str = f"{date_list[0]}.{date_list[1]}.{date_list[2]}"
```

```
    print(
```

```
        '| {:>4} | {:<30} | {:<20} | {:<15} |'.format(
```

```
            idx,
```

```
            contact.get('name', ""),
```

```
            contact.get('number', ""),
```

```
            date_str
```

```
)  
)  
print(line)
```

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

```
    count = 0
```

```
    parts = command.split(' ', maxsplit=1)
```

```
    num = parts[1]
```

```
    for contact in contacts:
```

```
        if contact.get('number') == num:
```

```
            count += 1
```

```
            print(contact.get('name', ""))
```

```
    if count == 0:
```

```
        print("Контакт не найден в телефонной книге.")
```

```
elif command == 'help':
```

```
    print("Список команд:\n")
```

```
    print("add - добавить контакт")
```

```
    print("list - вывести список контактов")
```

```
    print("select <номер> - найти контакт по номеру телефона")
```

```
    print("help - отобразить справку")
```

```
    print("exit - завершить работу с программой")
```

```
else:
```

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

```

>>> add
Фамилия и инициалы? K. F. J.
Номер телефона? 45678
Дата рождения (в формате XX.XX.XXXX)? 06.05.2001
>>> add
Фамилия и инициалы? YYYXxx.o
Номер телефона? 2434995
Дата рождения (в формате XX.XX.XXXX)? 05.05.2001
>>> add
Фамилия и инициалы? hjke.A
Номер телефона? 13459
Дата рождения (в формате XX.XX.XXXX)? 06.04.2001
>>> add
Фамилия и инициалы? rfgukA
Номер телефона? 84838383
Дата рождения (в формате XX.XX.XXXX)? 06.05.2000
>>> list
+-----+-----+-----+-----+
| № |          ФИО          | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
| 1 | rfgukA                | 84838383       | 6.5.2000      |
+-----+-----+-----+-----+
| 2 | hjke.A                | 13459          | 6.4.2001      |
+-----+-----+-----+-----+
| 3 | YYYXxx.o             | 2434995        | 5.5.2001      |
+-----+-----+-----+-----+
| 4 | K. F. J.              | 45678          | 6.5.2001      |
+-----+-----+-----+-----+
>>> select 84838383
rfgukA

```

Рис. 5 — Индивидуальное задание №2

5. Зафиксировали все изменения в репозиторий.

```

Asus@DESKTOP-KEJGOCU MINGW64 ~/gitprogect/Python_LAB_5 (main)
$ git commit -m"Indiv, prim"
[main 2fd4ade] Indiv, prim
17 files changed, 273 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/Python_LAB_5.iml
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 Indiv/indiv_1.py
create mode 100644 Indiv/indiv_2.py
create mode 100644 Primer/.idea/.gitignore
create mode 100644 Primer/.idea/Primer.iml
create mode 100644 Primer/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 Primer/.idea/misc.xml
create mode 100644 Primer/.idea/modules.xml
create mode 100644 Primer/pr_1.py
create mode 100644 Primer/pr_2.py

```

Рис. 16 — Фиксирование изменений

Контрольные вопросы:

1. Что такое множества в языке Python?

Множество — неупорядоченная коллекция уникальных элементов.

2. Как осуществляется создание множеств в Python?

`my_set = {1, 2, 3}` или `my_set = set([1, 2, 3])`

3. Как проверить присутствие/отсутствие элемента в множестве?

`if elem in my_set:` или `if elem not in my_set:`

4. Как выполнить перебор элементов множества?

`for item in my_set:`

`print(item)`

5. Что такое set comprehension?

Создание множества в одну строку: `{x**2 for x in range(5)}`

6. Как выполнить добавление элемента во множество?

`my_set.add(4)` - один элемент, `my_set.update([5, 6])` - несколько

7. Как выполнить удаление одного или всех элементов множества?

`my_set.remove(4)` - удалить элемент, `my_set.clear()` - очистить всё

8. Как выполняются основные операции над множествами?

Объединение: `set1 | set2` или `set1.union(set2)`

Пересечение: `set1 & set2` или `set1.intersection(set2)`

Разность: `set1 - set2` или `set1.difference(set2)`

9. Как определить надмножество или подмножество?

`set1.issuperset(set2)` - надмножество, `set1.issubset(set2)` -

подмножество

10. Каково назначение множеств `frozenset`?

Неизменяемое множество, можно использовать как ключ словаря

11. Как преобразовать множества?

`list(my_set)` - в список, `str(my_set)` - в строку, `dict(zip(my_set, values))` -

в словарь

12. Что такое словари в языке Python?

Неупорядоченная коллекция пар ключ-значение

13. Может ли функция `len()` быть использована со словарями?

Да, `len(my_dict)` возвращает количество пар ключ-значение

14. Какие методы обхода словарей известны?

`for key in my_dict:`

`for value in my_dict.values():`

`for key, value in my_dict.items():`

15. Как получить значения из словаря по ключу?

`my_dict[key]` или `my_dict.get(key)` (второй способ безопаснее)

16. Как установить значение в словаре по ключу?

`my_dict[key] = value` или `my_dict.update({key: value})`

17. Что такое словарь включений?

Создание словаря в одну строку: `{x: x**2 for x in range(5)}`

18. Функция `zip()`

Объединяет элементы нескольких последовательностей:

`names = ['a', 'b']`

`ages = [1, 2]`

`dict(zip(names, ages))` `# {'a': 1, 'b': 2}`

19. Модуль `datetime`

`datetime.now()` - текущая дата и время

`date.today()` - текущая дата

`timedelta` - разница между датами

Форматирование дат: `strftime()` и `strptime()`

Вывод: приобрели навыков по работе с множествами и словарями при написании программ на языке программирования Python версии 3.x.