

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

«Модули и пакеты»

**Отчет по лабораторной работе № 2.13
по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Гасанов Г. М. « » 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Выполните индивидуальные задания.

Задание 1. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Номер варианта уточнить у преподавателя.

Модуль:

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

def del_number(type='even'):
    def helper(lst):
        if type == 'even':
            lst = [i for i in lst if i % 2 != 0]
        else:
            lst = [i for i in lst if i % 2 == 0]
        return lst

    return helper
```

Программа:

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

from ind_del_number import del_number

if __name__ == '__main__':
    print(del_number(input('Введите even/odd: '))
          (map(int, input('Введите список целых чисел: ').split(', '))))
```

```
Введите even/odd: odd
Введите список целых чисел: 2, 3, 5, 1, 4
[2, 4]
```

Рисунок 8 – Результат работы программы

Задание 2. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `__all__` в файле `__init__.py` пакета.

Содержание пакета:

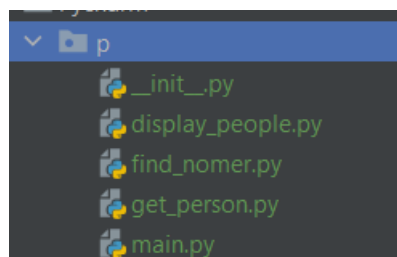


Рисунок 9 – Содержание проекта

`__init__.py`

```
__all__ = ["main", "display_people", "find_nomer", "get_person"]
```

`display_people.py`

```

def display_people(staff):
    """
    Список данных о людях.
    """
    if staff:
        # Заголовок таблицы.
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 14
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^14} |'.format(
                "№",
                "Фамилия Имя",
                "Номер телефона",
                "Дата рождения"
            )
        )
        print(line)

        # Вывести данные о всех людях.
        for idx, person in enumerate(staff, 1):
            print(
                f'| {idx:>4} |'
                f'| {person.get("name", ""):<30} |'
                f'| {person.get("number", 0):<20} |'
                f'| {person.get("birthday")} |'
            )
            print(line)

    else:
        print("Результат не найден")

```

find_nomer.py

```

def find_nomer(staff, nomer):
    """
    Выбрать людей с заданным номером телефона.
    """
    # Сформировать список людей.
    result = []

    for n in staff:
        if nomer in n.values():
            result.append(n)

    # Проверка на наличие записей
    if len(result) == 0:
        return None

    # Возвратить список выбранных людей.
    return result

```

get_person.py

```

from datetime import datetime

def get_person():
    """
    Запросить данные о человеке.
    """
    name = input("Фамилия Имя: ")
    number = int(input("Номер телефона: "))
    bday = list(map(int, input("Дата рождения: ").split('.')))
    d_bday = datetime(bday[2], bday[1], bday[0])

    # Создать словарь.
    return {
        'name': name,
        'number': number,
        'birthday': d_bday,
    }

```

main.py

```

import sys

from Pycharm.p.display_people import display_people
from Pycharm.p.find_nomer import find_nomer
from Pycharm.p.get_person import get_person

def main():
    """
    Главная функция программы.
    """
    # Список людей.
    people = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input("Введите команду >>> ").lower()

        # Выполнить действие в соответствии с командой.
        if command == 'exit':
            break

        elif command == 'add':
            # Запросить данные о человеке.
            person = get_person()

            # Добавить в словарь список.
            people.append(person)
            # Отсортировать список в случае необходимости.
            if len(people) > 1:
                people.sort(key=lambda item: item.get('d_bday', ''))

        elif command == 'list':
            # Отобразить всех людей.
            display_people(people)

        elif command == 'find':
            n = int(input('Введите номер телефона: '))

```

```

        # Выбрать людей с заданной фамилией.
        finded = find_nomer(people, n)
        # Отобразить выбранных работников.
        display_people(finded)

    elif command == 'help':
        # Вывести справку о работе с программой.
        print("Список команд:\n")
        "add - добавить человека;\n"
        "list - вывести список людей;\n"
        "find - найти человека по фамилии;\n"
        "help - отобразить справку;\n"
        "exit - завершить работу с программой.\n")

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

```

Код основной программы:

```

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

from p.main import main

if __name__ == '__main__':
    main()

```

```
C:\Users\dimu7\AppData\Local\Programs\Python\Python311\python.exe C:\Users\dimu7\Desktop\0111\Laba_16
Введите команду >>> add
Фамилия Имя: gasanov gamid
Номер телефона: 3
Дата рождения: 18.08.2004
Введите команду >>> add
Фамилия Имя: dsds dsds
Номер телефона: 4
Дата рождения: 15.09.2000
Введите команду >>> list
+-----+-----+-----+-----+
| № | Фамилия Имя | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
| 1 | gasanov gamid | 3 | 2004-08-18 00:00:00 |
| 2 | dsds dsds | 4 | 2000-09-15 00:00:00 |
+-----+-----+-----+-----+
Введите команду >>> find
Введите номер телефона: 4
+-----+-----+-----+-----+
| № | Фамилия Имя | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
| 1 | dsds dsds | 4 | 2000-09-15 00:00:00 |
+-----+-----+-----+-----+
Введите команду >>> help
Список команд:
add - добавить человека;
list - вывести список людей;
find - найти человека по номеру телефона;
help - отобразить справку;
exit - завершить работу с программой.
```

Рисунок 8 – Результат работы программы

Вопросы для защиты работы

1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением .py. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы: программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы. Стоит заметить, что модули могут быть написаны не только на языке Python, но и на других языках (например C).

2. Какие существуют способы подключения модулей в языке Python?

Самый простой способ импортировать модуль в Python это воспользоваться конструкцией:

```
import имя_модуля
```

За один раз можно импортировать сразу несколько модулей, для этого их нужно перечислить через запятую после слова `import`. Если вы хотите задать псевдоним для модуля в вашей программе, можно воспользоваться вот таким синтаксисом:

```
import имя_модуля as новое_имя
```

Для импортирования нескольких функций из модуля, можно перечислить их имена через запятую `from имя_модуля import имя_объекта1, имя_объекта2`

3. Что является пакетом языка Python?

Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py`. Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла `__init__.py`?

В `__init__.py` файл заставляет Python рассматривать каталоги, содержащие его, как модули. Кроме того, это первый файл, загружаемый в модуль, поэтому вы можете использовать его для выполнения кода, который хотите запускать каждый раз при загрузке модуля, или для указания экспортируемых подмодулей.

5. Каково назначение переменной `__all__` файла `__init__.py`

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию `from имя_пакета import *`

Вывод: в ходе выполнения практической работы были приобретены навыки по работе декораторами функций при написании программ с помощью языка программирования Python.