

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.13
дисциплины «Программирование на Python»
Вариант ____

Выполнил:
Иващенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и
вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики:
Воронкин Роман Александрович,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: «Модули и пакеты»

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

Порядок выполнения работы

Индивидуальное задание 1. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключён в основную программу с помощью одного из вариантов команды `import`.

Таблица 1.1 – Код программы `individual_1.py`

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

from type_transform_module import type_transform

if __name__ == "__main__":
    selected_type = type_transform(input("Введите тип (tuple/list): "))
    collection = selected_type(input("Введите числа через пробел: ").split())
    print(f"Результат: {collection}")
    print(f"Выбранный тип - {type(collection)}")
```

Таблица 1.2 – Код пакета `type_transform_module.py`

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

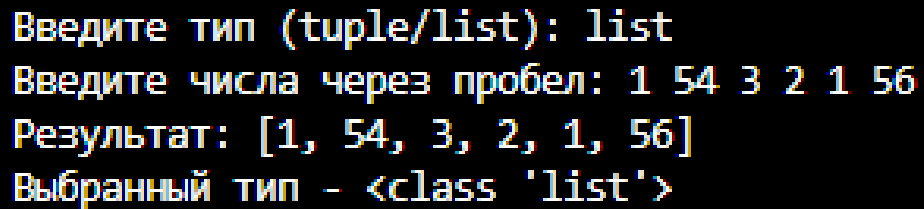
def type_transform(selected_type):
    """
    Функция принимает строковое значение. Рекомендуется
    использовать значения 'tuple' или 'list'
    """

    def transform(numbers):
        """
        Функция принимает числовые значения и, в зависимости от
        указанного строкового значения внешней функции, преобразует
        введённые значения в кортеж или список. Возвращает преобразованный
        результат
        """

        collection = [int(value) for value in numbers]
```

```
if selected_type == 'tuple':
    return tuple(collection)
elif selected_type == 'list':
    return list(collection)

return transform
```



```
Введите тип (tuple/list): list
Введите числа через пробел: 1 54 3 2 1 56
Результат: [1, 54, 3, 2, 1, 56]
Выбранный тип - <class 'list'>
```

Рисунок 1 – Результат выполнения программы individual_1.py

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

Таблица 2.1 – Код программы individual_2.py

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

from notes import *

if __name__ == "__main__":
    """
    Основная программа
    """

    member_list = []

    while True:
        cmd = input(">>> ")

        if cmd == "help":
            help.print_help()

        elif cmd == "add":
            member_list.append(add.add())
```

```

        member_list.sort(key=lambda item: item.get('phone')[:3])

    elif cmd == "list":
        print_list.print_list(member_list)

    elif cmd == "find":
        find.find_member(input("Введите фамилию: "), member_list)

    elif cmd == "exit":
        print("Завершение работы программы...")
        break

    else:
        print(f"Команды {cmd} не существует")

```

Таблица 2.2 – Код файла __init__.py

```

__all__ = [
    "find",
    "print_list",
    "add",
    "help"
]

```

Таблица 2.3 – Код пакета find.py

```

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

def find_member(surname, member_list):
    """
    Функция для вывода на экран всех записей, чьи фамилии совпадают
    с введённой (не возвращает никаких значений)
    """

    count = 0

    for member in member_list:
        if member['surname'] == surname:
            print(f"{member['surname']} {member['name']}, "
                  f"{member['phone']}, {member['date']}")
            count += 1

    if count == 0:
        print("Записи не найдены")

```

Таблица 2.4 – Код пакета print_list.py

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

def print_list(list):
    """
    Функция выводит на экран список всех существующих записей
    """

    for member in list:
        print(f"{member['surname']} {member['name']}, "
              f"{member['phone']}, {member['date']}")
```

Таблица 2.5 – Код пакета add.py

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

def add():
    """
    Функция добавления новой записи, возвращает запись
    """

    surname = input("Введите фамилию: ")
    name = input("Введите имя: ")
    phone = input("Введите номер телефона: ")
    date = tuple(map(int, input("Введите дату рождения: ").split('.')))

    new_member = {'surname': surname,
                  'name': name,
                  'phone': phone,
                  'date': date
                 }

    return new_member
```

Таблица 2.6 – Код программы help.py

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

def print_help():
    """
    Функция вывода доступных пользователю команд
    """

    print("list - вывод всех добавленных записей")
    print("add - добавление новых записей")
```

```
print("find - найти запись по фамилии")
print("exit - завершение работы программы")
```

```
>>> help
list - вывод всех добавленных записей
add - добавление новых записей
find - найти запись по фамилии
exit - завершение работы программы
>>> add
Введите фамилию: Иванов
Введите имя: Иван
Введите номер телефона: 78889995623
Введите дату рождения: 01.01.2000
>>> list
Иванов Иван, 78889995623, (1, 1, 2000)
>>> find
Введите фамилию: Иванов
Иванов Иван, 78889995623, (1, 1, 2000)
>>> □
```

Рисунок 2 – Результат выполнения программы individual_2.py

Контрольные вопросы

1. Что является модулем языка Python?

В Python модуль – это файл с расширением «.py», который содержит код на языке Python. Модуль может содержать функции, переменные и другие объекты, и его можно использовать для организации кода в более крупных проектах.

2. Какие существуют способы подключения модулей в языке Python?

Для подключения модулей в Python можно использовать ключевые слова «import» и «from». Например:

```
import module_name
from module_name import func
```

3. Что является пакетом языка Python?

Пакет в Python – это способ организации модулей в иерархическую структуру. Пакет представляет собой каталог, содержащий специальный файл «__init__.py» и может содержать другие модули или подпакеты.

4. Каково назначение файла __init__.py?

Файл «__init__.py» в пакете Python выполняет роль инициализации для этого пакета. Он может содержать код, который будет выполнен при импорте пакета, и он обозначает, что каталог является пакетом.

5. Каково назначение переменной __all__ файла __init__.py?

Переменная «__all__» в файле «__init__.py» пакета используется для определения списка модулей, которые будут импортированы при использовании конструкции «from package import *». Это позволяет контролировать, какие имена экспортируются из пакета при использовании подобного импорта.

Выводы: В процессе выполнения лабораторной работы были приобретены навыки по работе с модулями и пакетами языка программирования Python версии 3.x, были выполнены две индивидуальные задачи, в рамках которых индивидуальные программы из лабораторных работ 2.11 и 2.8 были разбиты на модули и текущей и подключены к основной исполняемой программе.