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

Коновалова В.Н. « » 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.

Ход 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.

5. Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве параметров фамилию и имя, а затем, заносит в шаблон эти данные. Сам шаблон – это строка, которая передается внешней функции и, например, может иметь такой вид: «Уважаемый %F%, %N%! Вы делаете работу по замыканиям функций.» Здесь %F% - это фрагмент куда нужно подставить фамилию, а %N% - фрагмент, куда нужно подставить имя. (Шаблон может быть и другим, вы это определяете сами). Здесь важно, чтобы внутренняя функция умела подставлять данные в шаблон, формировать новую строку и возвращать результат. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

Модуль:

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

def sample(string):
    def name_surname(n, s):
        sample_data = string.replace("%N%", n)
        sample_data = sample_data.replace("%F%", s)
        return sample_data
```

```
    return name_surname
```

Основная программа:

```
from module_1 import sample

if __name__ == "__main__":
    sample_string = (
        "Уважаемый(ая) %F% %N%! Вы делаете работу по замыканиям функций."
    )
    name, surname = input("Введите имя и фамилию: ").split()
    print(sample(sample_string)(name, surname))
```

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. Номер варианта уточнить у преподавателя.

8. Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

Код:

__init__.py

```
__all__ = ["add", "listt", "select_train", "main"]
```

add.py

```
def add(trains, name, num, time):
    """
        Добавляем маршруты
    """
    # Создать словарь
    train = {
        'name': name,
        'num': num,
        'time': time,
    }

    trains.append(train)
    if len(trains) > 1:
        trains.sort(key=lambda item: item.get('num', ''))
```

listt.py

```
def listt(trains):
    """
        Отобразить список маршрутов
    """
    if trains:
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 17
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^17} |'.format(
                "№",
                "Пункт назначения",
                "Номер поезда",
                "Время отправления"
            )
        )
        print(line)
        # Вывести данные о всех маршрутах
        for idx, train in enumerate(trains, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:>17} |'.format(
                    idx,
                    train.get('name', ''),
                    train.get('num', ''),
                    train.get('time', 0)
                )
            )

        print(line)
    else:
        print("Список маршрутов пуст!")
```

select_train.py

```
def select_train(trains, number):
    """""""
    Выбрать маршруты с заданным номер поезда
    """""""
    # Сформировать список поездов
    result = []
    for rattler in trains:
        if rattler.get('num') == number:
            result.append(rattler)

    # Возвратить список выбранных маршрутов
    return result
```

main.py

```
from packet.add import add
from packet.listt import listt
from packet.select_train import select_train

import sys
```

```
def main():
    """
        Главная функция программы
    """
    # Сформировать список маршрутов
    route = []

    # Организовать бесконечный цикл запроса команд
    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break

        elif command == 'add':
            name = input("Название пункта назначения: ")
            num = int(input("Номер поезда: "))
            time = input("Время отправления: ")

            add(route, name, num, time)

        elif command == 'list':
            listt(route)

        elif command.startswith('select '):
            parts = command.split(' ', maxsplit=1)
            number = int(parts[1])
            selected = select_train(route, number)
            listt(selected)

        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить поезд;")
            print("list - вывести список поездов;")
            print("select <номер поезда> - запросить информацию о выбранном
поезде;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")

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

Код основной программы:

```
from packet.main import main

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

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 имортирования 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.