

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Лабораторная работа 5

Аннотация типов

Выполнил студент группы ИВТ-б-о-20-1

Симанский М.Ю « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

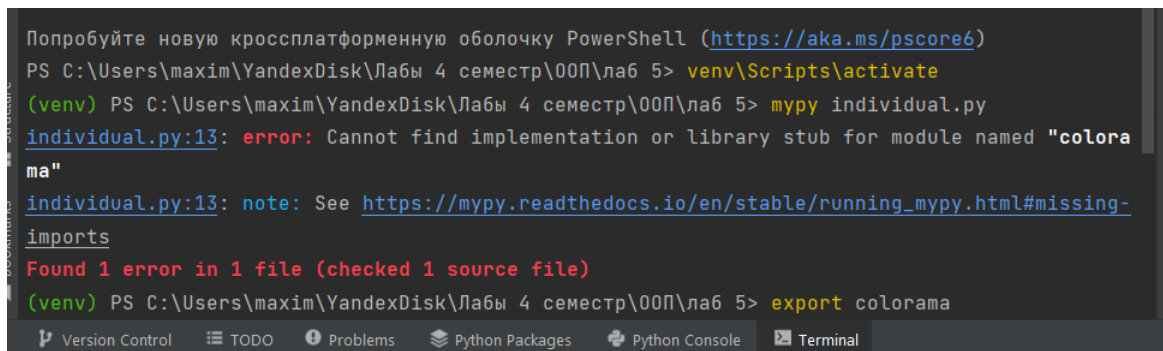
Цель работы: приобретение навыков по работе с аннотациями типов при написании программ с помощью языка программирования Python версии 3.x. Рассмотрен вопрос контроля типов переменных и функций с использованием комментариев и аннотаций. Приведено описание PEP'ов, регламентирующих работу с аннотациями, и представлены примеры работы с инструментом туру для анализа Python кода.

Индивидуальное задание

Выполнить индивидуальное задание 2 лабораторной работы 2.19, добавив аннотации типов (рис. 1). Выполнить проверку программы с помощью утилиты туру(рис. 2).

```
def tree(directory: Path) -> None:
    print(Fore.RED + f' / {directory}')
    for path in sorted(directory.rglob('*')):
        depth: int = len(path.relative_to(directory).parts)
        spacer: str = ' ' * depth
        if path.suffix == ".exe":
            print(Fore.BLUE + f'{spacer} ++ {path.name}')
        if path.suffix == ".py":
            print(Fore.LIGHTGREEN_EX + f'{spacer} ++ {path.name}')
        if path.suffix == (".txt" or ".html"):
            print(Fore.WHITE + f'{spacer} ++ {path.name}')
```

Рисунок 1 – Пример аннотации типа



```
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)
PS C:\Users\maxim\YandexDisk\Лабы 4 семестр\00П\лаб 5> venv\Scripts\activate
(venv) PS C:\Users\maxim\YandexDisk\Лабы 4 семестр\00П\лаб 5> mypy individual.py
individual.py:13: error: Cannot find implementation or library stub for module named "colora
ma"
individual.py:13: note: See https://mypy.readthedocs.io/en/stable/running\_mypy.html#missing-
imports
Found 1 error in 1 file (checked 1 source file)
(venv) PS C:\Users\maxim\YandexDisk\Лабы 4 семестр\00П\лаб 5> export colorama
```

Рисунок 2 – Проверка туру

Ответы на контрольные вопросы

1. Как осуществляется объявление класса в языке Python? - Классы объявляются с помощью ключевого слова `class` и имени класса.
2. Чем атрибуты класса отличаются от атрибутов экземпляра? - Атрибуты класса определены внутри класса, но вне каких-либо методов. Их значения одинаковы для всех экземпляров этого класса. Так что вы можете рассматривать их как тип значений по умолчанию для всех наших объектов. Что касается переменных экземпляра, они хранят данные, уникальные для каждого объекта класса. В этой теме мы рассмотрим только атрибуты класса, но не волнуйтесь, у вас будет достаточно времени, чтобы узнать больше и об атрибутах экземпляра.
3. Каково назначение методов класса? - Методы определяют функциональность объектов, принадлежащих конкретному классу.
4. Для чего предназначен метод `__init__()` класса? – Для объявления конструктора класса.
5. Каково назначение `self` ? – Указание объекта на самого себя.
6. Как добавить атрибуты в класс? - Атрибуты экземпляра - это как раз те, которые мы определяем в методах, поэтому по определению мы можем создавать новые атрибуты внутри наших пользовательских методов.

Как осуществляется управление доступом к методам и атрибутам в языке Python? - т.д.) - В Python таких возможностей нет, и любой может обратиться к атрибутам и методам вашего класса, если возникнет такая необходимость. Это существенный недостаток этого языка, т.к. нарушается один из ключевых принципов ООП – инкапсуляция. Хорошим тоном считается, что для чтения/изменения какого-то атрибута должны использоваться специальные методы, которые называются `getter/setter`, их можно реализовать, но ничего не мешает изменить атрибут напрямую. При этом есть соглашение, что метод или атрибут, который начинается с нижнего подчеркивания, является скрытым, и снаружи класса трогать его

не нужно (хотя сделать это можно).

7. В Python есть встроенная функция `instance ()`, которая сравнивает значение с указанным типом. Если данное значение и тип соответствуют, он вернет `true`, иначе `false`. Используя `isinstance ()`, вы можете проверить строку, число с плавающей точкой, `int`, список, кортеж, `dict`, `set`, `class` и т. Д.