

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №4.5

Аннотация типов

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе с аннотациями 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 mypy для анализа Python кода.

Ход работы: Примеры

<https://github.com/Mirror-Shard/L4.5>

1. Изучил теоретический материал и приступил к выполнению примеров:

```
(tools) C:\Users\1\Desktop\Алгоритмизация\Lab12\L2.17\examples>python example_1.py add workers.json --name="anton" --post="president" --year=2002
```

Рисунок 1 – Добавление работника в новый файл

```
(tools) C:\Users\1\Desktop\Алгоритмизация\Lab12\L2.17\examples>python example_1.py display workers.json
```

№	Ф.И.О.	Должность	Год
1	anton	president	2002

Рисунок 2 – Вывод информации о работниках из файла

Индивидуальное задание 1

Выполнить индивидуальное задание 2 лабораторной работы 2.19, добавив аннотации типов. Выполнить проверку программы с помощью утилиты mypy.

1. Добавил аннотации в код основной функции своей программы:

```
def tree(directory: Path, dir: bool, pattern: str, sep: str) -> None:
    if dir:
        pattern: str = '.'
    for path in sorted(directory.rglob(pattern)):
        depth: int = len(path.relative_to(directory).parts)
        spacer: str = '\t' * depth
        print(spacer + sep + ' ' + path.name)
```

Рисунок 3 – Добавление аннотаций

2. Затем ввёл стандартную команду, программа успешно работает:

```
(base) C:\Users\1\Desktop\00П\Лабораторная 4.5\Lab_4.5\индивидуализация\Лабораторная 2.19\L2.19\examples"
|-- data.json
|-- doc
    |-- inspectionProfiles
        |-- profiles_settings.xml
        |-- Project_Default.xml
    |-- Отчёт №2.17.pdf
|-- in.py
|-- ind_1.py
|-- ind_2.py
|-- new
    |-- some.txt
|-- s.py
```

Рисунок 4 – Работа программы

Контрольные вопросы:

1. Для чего нужны аннотации типов в языке Python? Для того чтобы повысить информативность исходного кода, и иметь возможность с помощью сторонних инструментов производить его анализ.

2. Как осуществляется контроль типов в языке Python? Один из возможных вариантов (наверное самый логичный) решения данной задачи — это использование комментариев, составленных определенным образом.

3. Какие существуют предложения по усовершенствованию Python для работы с аннотациями типов?

- PEP 3107;
- PEP 484;
- PEP 526;
- PEP 563.

4. Как осуществляется аннотирование параметров и возвращаемых значений функций?

Аннотация для аргумента определяется через двоеточие после его

имени. Аннотация, определяющая тип возвращаемого функцией значения, указывается после ее имени с использованием символов ->

5. Как выполнить доступ к аннотациям функций?

Доступ к использованным в функции аннотациям можно получить через атрибут `__annotations__`, в котором аннотации представлены в виде словаря, где ключами являются атрибуты, а значениями – аннотации.

6. Как осуществляется аннотирование переменных в языке Python?

```
var = value # type: annotation
```

```
var: annotation; var = value
```

```
var: annotation = value
```

7. Для чего нужна отложенная аннотация в языке Python?

Чтобы убрать ошибку, возникающую во время импорта модуля. Вывод: в ходе выполнения лабораторной работы были приобретены основные навыки по работе с аннотацией типов в языке программирования Python.

Вывод: в ходе работы приобрёл навыки по работе с аннотациями с помощью языка программирования Python версии 3. 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 `mypy` для анализа Python кода.