

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение высшего образования**
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ

по лабораторной работе №4.5

Дисциплина: «Объектно-ориентированное программирование»

Тема: «Аннотации типов»

Вариант 5

Выполнила: студентка 3 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2023

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

Практическая часть:

1. Создала репозиторий, клонировала его, изменила файл .gitignore.
2. Проработала пример.

```
>>> load work.txt
>>> list
```

--+			
№	Ф.И.О.	Должность	Год
+-----+			
1	Диченко Д.А.	стажер	2020
2	Иванов И.И.	бухгалтер	2012

Рисунок 1. Выполнение примера

3. Выполнила индивидуальное задание.

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

```
Введите команду ('help' для списка команд): help
all - просмотр полного пути файла
files - просмотр всех файлов в директории
size - просмотр размера файла
save - сохранение данных в json-файл
mkdir - создание директории
rmdir - удаление директории
mk - создание файла
rm - удаление файла
Введите команду ('help' для списка команд): all
C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗ\Воронкин х3\Lab_4_5\prog
Введите команду ('help' для списка команд): files
>> individual.py
>> prim.py
>> work.txt
>> workers4.log
>> individual.py
>> prim.py
>> work.txt
>> workers4.log
Введите команду ('help' для списка команд): mkdir New
>> individual.py
>> New
>> prim.py
>> work.txt
>> workers4.log
```

Рисунок 2. Выполнение индивидуального задания

```
C:\Users\super\OneDrive\Рабочий стол\DI\ВУЗЬ\Воронкин х3\Lab_4_5\prog>mypy individual.py
Success: no issues found in 1 source file
```

Рисунок 3. Проверка программы с помощью mypy

Контрольные вопросы:

1 Для чего нужны аннотации типов в языке Python?

Для начала ответим на вопрос: зачем нужны аннотации в Python? Если кратко, то ответ будет таким: для того чтобы повысить информативность исходного кода, и иметь возможность с помощью сторонних инструментов производить его анализ. Одной из наиболее востребованных, в этом смысле, тем является контроль типов переменных. Несмотря на то, что Python – это язык с динамической типизацией, иногда возникает необходимость в контроле типов.

Согласно PEP 3107 могут быть следующие варианты использования аннотаций:

- проверка типов;
- расширение функционала IDE в части предоставления информации об ожидаемых типах
 - аргументов и типе возвращаемого значения у функций;
 - перегрузка функций и работа с дженериками;
 - взаимодействие с другими языками;
 - использование в предикатных логических функциях;
 - маппинг запросов в базах данных;
 - маршалинг параметров в RPC (удаленный вызов процедур).

2 Как осуществляется контроль типов в языке Python?

- Использование комментариев, составленных определенным образом.

Например: `name = "John" # type: str`

- Если необходимо проконтролировать, что переменной `name` будут присваиваться значения только строкового типа, мы должны: во-первых указать в комментарии о нашем намерении – это мы сделали, во-вторых

использовать специальный инструмент, который выполнит соответствующую проверку. Таким инструментом является `mypy`.

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

4 Как осуществляется аннотирование параметров и возвращаемых значений функций?

В функциях мы можем аннотировать аргументы и возвращаемое значение. Выглядеть это может так.

```
def repeater(s: str, n: int) -> str:  
    return s * n
```

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. Возвращаемое функцией значение хранится в записи с ключом `return`.

6 Как осуществляется аннотирование переменных в языке Python?

Можно использовать один из трех способов создания аннотированных переменных.

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

7 Для чего нужна отложенная аннотация в языке Python?

Чтобы определить тип в аннотации для переменной, которая еще не определена.

Вывод: в результате выполнения работы были приобретены навыки 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 РЕР'ов, регламентирующих работу с аннотациями, и представлены примеры работы с инструментом муру для анализа Python кода.