

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии  
Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**  
дисциплины «Объектно-ориентированное программирование»  
Вариант 3

Выполнил:  
Говоров Егор Юрьевич  
3 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»,  
очная форма обучения

---

(подпись)

Проверил:  
Воронкин Роман Александрович,  
доцент департамента цифровых,  
робототехнических систем и  
электроники

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

## Тема: Аннотация типов

Цель: 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 муру для анализа Python кода.

Ссылка на репозиторий: [https://github.com/Artorias1469/object-oriented-programming\\_5.git](https://github.com/Artorias1469/object-oriented-programming_5.git)

### Ход работы:

#### 1. Проработка примера:

```
D:\Users\Admin\anaconda3\envs\object-oriented-programming_5\python.exe "D:\Us
>>> add
Фамилия и инициалы? Говоров Егор Юрьевич
Должность? Руководитель
Год поступления? 2014
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Говоров Егор Юрьевич    |      Руководитель   |      2014     |
+-----+-----+-----+-----+
>>> save primer.xml
```

Рис 1. Результат выполнения примера

#### 2. Выполнение индивидуального задания:

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

```
D:\Users\Admin\anaconda3\envs\object-oriented-programming_5\python.exe
.
├── .mypy_cache
│   └── 3.11
│       ├── collections
│       ├── email
│       ├── importlib
│       ├── metadata
│       └── resources
├── os
├── sys
├── _typeshed
└── Primer
```

Рис 2. Результат работы из индивидуального задания по добавлению аннотации

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_5\Prog>mypy Ind_1.py  
Success: no issues found in 1 source file
```

Рис 3. Проверка типов данных в коде с помощью утилиты mypy

### Контрольные вопросы

#### 1. Для чего нужны аннотации типов в языке Python?

Аннотации типов в Python служат для указания ожидаемых типов данных для параметров функций и возвращаемых значений. Это помогает улучшить читаемость кода и облегчает его поддержку, позволяя разработчикам быстрее понимать, какие типы данных используются в функции.

#### 2. Как осуществляется контроль типов в языке Python?

Контроль типов в Python осуществляется неявно, так как язык является динамически типизированным. Это означает, что типы переменных проверяются во время выполнения, а не на этапе компиляции.

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

Существуют различные предложения по усовершенствованию Python, включая улучшение поддержки отложенных аннотаций и расширение возможностей модуля typing

#### 4. Как осуществляется аннотирование параметров и возвращаемых значений функций?

Аннотирование параметров и возвращаемых значений функций осуществляется с помощью синтаксиса, который включает указание типа после имени параметра и после стрелки `->` для возвращаемого значения. Например: `def add(a: int, b: int) -> int: return a + b`

#### 5. Как выполнить доступ к аннотациям функций?

Доступ к аннотациям функций можно получить через атрибут `__annotations__`. Этот атрибут возвращает словарь, где ключами являются имена параметров, а значениями — аннотированные типы.

#### 6. Как осуществляется аннотирование переменных в языке Python?

Аннотирование переменных в Python можно осуществить с помощью синтаксиса, аналогичного аннотированию параметров функций. Например: `x: int = 10, name: str = "Alice"`

#### 7. Для чего нужна отложенная аннотация в языке Python?

Отложенная аннотация в Python позволяет ссылаться на типы, которые еще не определены, что особенно полезно в случаях, когда типы ссылаются друг на друга. Это достигается с помощью строки, содержащей имя типа, вместо непосредственного указания типа. с

Вывод: в ходе выполнения работы были приобретены навыки по работе аннотациями типов при написании программ с помощью языка программирования Python версии 3.x., был рассмотрен вопрос контроля типов переменных и функций с использованием комментариев и аннотаций, приведено описание PEP, регламентирующих работу с аннотациями, и представлены примеры работы с инструментом `mypy` для анализа Python кода.