

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №28
дисциплины «Основы программной инженерии»

Выполнил:
Матвеев Александр Иванович
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Аннотация типов в Python.

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.

Ход работы.

1. Создание нового репозитория с лицензией MIT.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * SashkaHacker / **Repository name *** laba10
✔ laba10 is available.

Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: Python
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: MIT License
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

Create repository

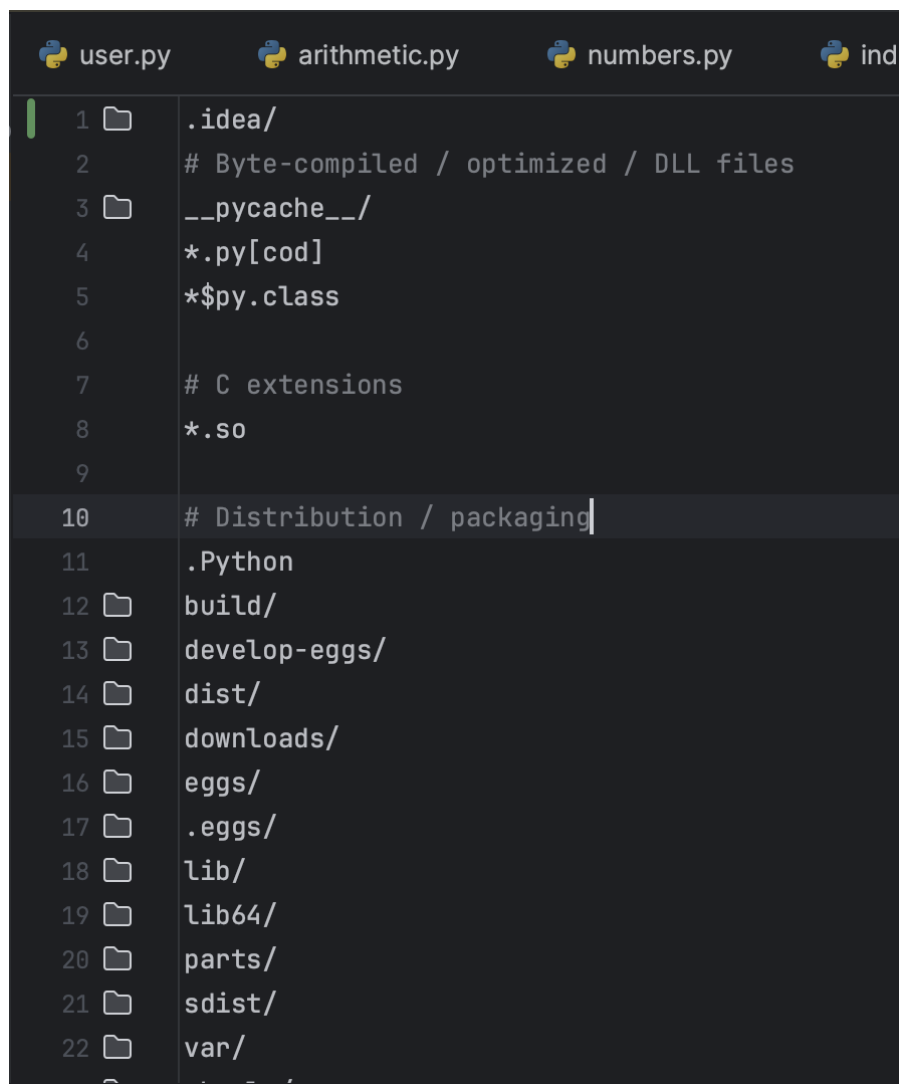
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
[→ GitHub git clone https://github.com/SashkaHacker/laba16.git
Cloning into 'laba16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.



```
user.py arithmetic.py numbers.py ind
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

Рисунок 3 – Файл .gitignore

4. Выполнение индивидуального задания №1.

```
(env) PS C:\Users\Sashka\Documents\GitHub\laba28\individuals> python .\individual1.py add -n Александр -s Матвеев -p 89614776749 -d 28:05:2004 data.json
(env) PS C:\Users\Sashka\Documents\GitHub\laba28\individuals> python .\individual1.py add -n Василий -s Булгаков -p 896925467365 -d 30:05:2000 data.json
(env) PS C:\Users\Sashka\Documents\GitHub\laba28\individuals> python .\individual1.py display data.json
```

№	Фамилия	Имя	Номер телефона	Дата рождения
1	Булгаков	Василий	896925467365	30:05:2000
2	Матвеев	Александр	89614776749	28:05:2004

Рисунок 4 – Демонстрация работы программы

```
(env) PS C:\Users\Sashka\Documents\GitHub\laba28\individuals> mypy .\individual1.py
Success: no issues found in 1 source file
```

Рисунок 5 – Проверка программы при помощи туру

Контрольные вопросы:

1. Аннотации типов в Python используются для указания ожидаемых типов переменных, аргументов функций и возвращаемых значений. Они помогают улучшить читаемость кода, облегчают рефакторинг, а также позволяют статическим анализаторам кода, таким как `mypy`, проверять код на соответствие указанным типам, что способствует обнаружению ошибок на ранних этапах разработки.

2. Контроль типов в Python осуществляется динамически во время выполнения программы. Однако с помощью аннотаций типов и статических анализаторов, таких как `mypy`, можно осуществлять статическую проверку типов до запуска программы.

3. Существует несколько предложений по усовершенствованию Python для работы с аннотациями типов, изложенных в документах PEP (Python Enhancement Proposals). Например, PEP 484 вводит синтаксис для аннотаций типов, PEP 526 добавляет синтаксис для аннотаций переменных, PEP 563 предлагает отложенное выполнение аннотаций типов.

4. Аннотирование параметров и возвращаемых значений функций осуществляется с помощью синтаксиса аннотаций, где после параметра ставится двоеточие и указывается тип, а после объявления функции ставится стрелка `->` и указывается тип возвращаемого значения. Пример:

5. Доступ к аннотациям функций можно получить через атрибут `__annotations__` функции.

6. Аннотирование переменных осуществляется с помощью синтаксиса аннотаций, где после имени переменной ставится двоеточие и указывается тип.

7. Отложенная аннотация (deferred annotation) в Python нужна для решения проблем, связанных с циклическими импортами и использованием типов, которые еще не были объявлены в момент аннотирования. Это достигается с помощью строки `from __future__ import annotations`, которая

говорит интерпретатору не выполнять аннотации немедленно, а отложить их до фактического использования.