

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 №29
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: Классы данных в Python.

Цель работы: приобретение навыков по работе с классами данных при написании программ с помощью языка программирования Python версии 3.x.

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

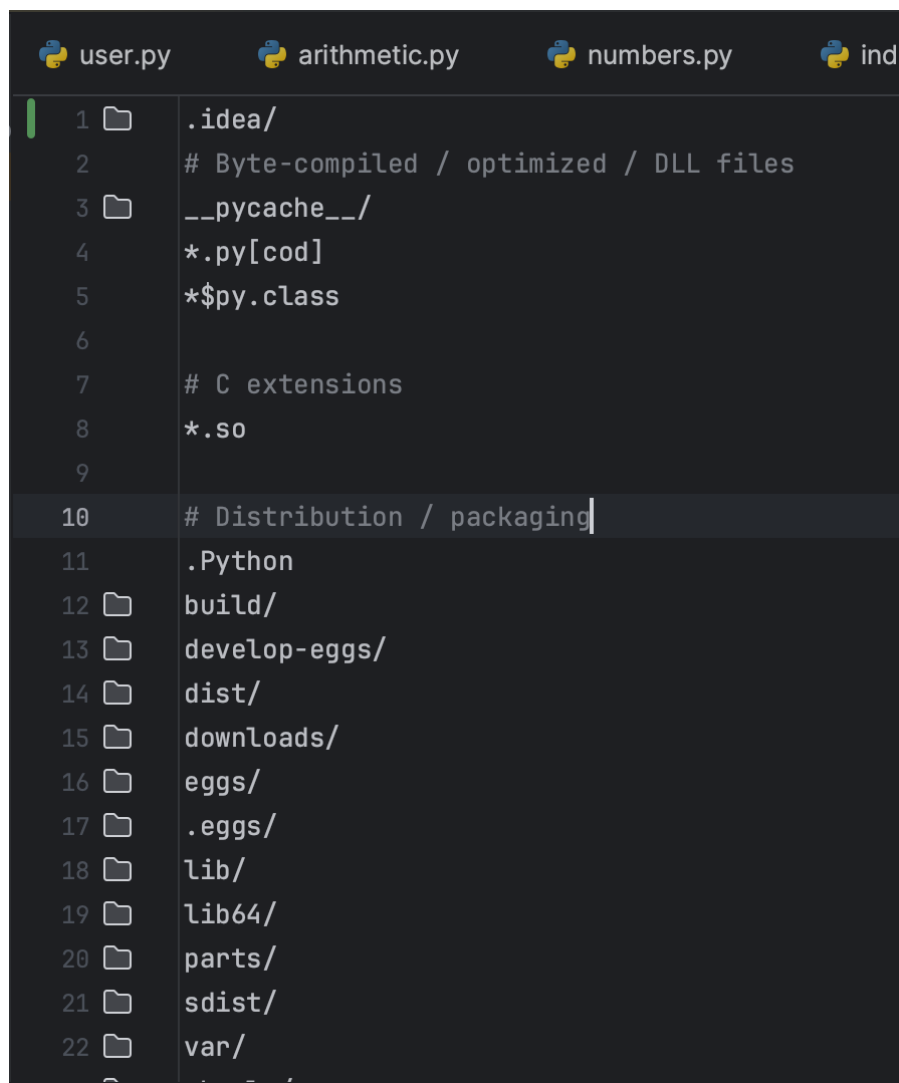
Рисунок 1 – Создание репозитория

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.



The screenshot shows a code editor with a dark theme. At the top, there are four tabs: 'user.py', 'arithmetic.py', 'numbers.py', and 'ind'. The active tab is 'user.py'. The editor displays the content of the '.gitignore' file, which is as follows:

```
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
12 .Python
13 build/
14 develop-eggs/
15 dist/
16 downloads/
17 eggs/
18 .eggs/
19 lib/
20 lib64/
21 parts/
22 sdist/
23 var/
```

Рисунок 3 – Файл .gitignore

4. Выполнение примера из лабораторной работы.

```

>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
load <имя_файла> - загрузить данные из файла;
save <имя_файла> - сохранить данные в файл;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? Матвеев А.И.
Должность? Главный
Год поступления? 2024
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|   1 | Матвеев А.И.            |      Главный        |      2024     |
+-----+-----+-----+-----+

```

Рисунок 4 – Демонстрация работы примера №1

5. Выполнение индивидуального задания.

```

(venv) → individual git:(develop) × python3 individual1.py add -n=Николай -s=Булавин -p=896444523429 -d=28:05:2000 data.xml
(venv) → individual git:(develop) × python3 individual1.py display data.xml
+-----+-----+-----+-----+
| № |          Фамилия          |      Имя      | Номер телефона | Дата рождения |
+-----+-----+-----+-----+
|   1 | Булавин                  | Николай      | 896444523429 | 28:05:2000 |
|   2 | Матвеев                  | Александр    | 89614776749  | 28:05:2004 |
|   3 | Алексеев                 | Сергей       | 89614773429  | 28:05:2020 |
+-----+-----+-----+-----+
(venv) → individual git:(develop) × python3 individual1.py select -p=89614776749 data.xml
Фамилия: Матвеев
Имя: Александр
Номер телефона: 89614776749
Дата рождения: 28:05:2004
(venv) → individual git:(develop) × 

```

Рисунок 4 – Демонстрация работы программы

Контрольные вопросы:

1. Для создания класса данных в Python используется декоратор `@dataclass` из модуля `dataclasses`.
2. По умолчанию декоратор генерирует: `__repr__`, `__init__`, `__eq__`.
3. Чтобы создать неизменяемый класс данных, нужно добавить параметр `frozen=True` в декоратор `@dataclass`.