

## Лабораторная работа 4.6


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.

Ссылка на GitHub: [https://github.com/RomanGorchakov/Py3\\_7](https://github.com/RomanGorchakov/Py3_7)

### Порядок выполнения работы

1. Создаём новый общедоступный репозиторий, в котором будет использована лицензия MIT и язык программирования Python.

 Start writing code ...

**Start a new repository for RomanGorchakov**

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name \*

Py3\_7

✔ Py3\_7 is available.

☒ **Public**  
Anyone on the internet can see this repository

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

Create a new repository

**Introduce yourself with a profile README**

Share information about yourself by creating a profile README, which appears at the top of your profile page.

RomanGorchakov / README.md **Create**

1 - 👋 Hi, I'm @RomanGorchakov

2 - 👀 I'm interested in ...

3 - 🌱 I'm currently learning ...

4 - ❤️ I'm looking to collaborate on ...

5 - 💬 How to reach me ...

6 - 🗨️ Pronouns: ...

7 - ⚡ Fun fact: ...

8

**Add a license to your project**

License	Permissions	Limitations	Conditions
Apache License 2.0			
GNU General Public License v3.0			
<b>MIT License</b>	<ul style="list-style-type: none"> <li>✓ Commercial use</li> <li>✓ Modification</li> <li>✓ Distribution</li> <li>✓ Private use</li> </ul>	<ul style="list-style-type: none"> <li>✗ Liability</li> <li>✗ Warranty</li> </ul>	<ul style="list-style-type: none"> <li>④ License and copyright notice</li> </ul>
BSD 2-Clause "Simplified" License			
BSD 3-Clause "New" or "Revised" License			
Boost Software License 1.0			
Creative Commons Zero v1.0 Universal			
Eclipse Public License 2.0			
GNU Affero General Public License v3.0			
GNU General Public License v2.0			

**MIT License**

Copyright (c) 2024 Roman Gorchakov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

To adopt MIT License, enter your details. You'll have a chance to review before committing a LICENSE file to a new branch or the root of your project.

**Year** ④  
2024

**Full name** ④  
RomanGorchakov

**Review and submit**

2. Теперь необходимо дополнить файл `.gitignore` с необходимыми правилами для языка программирования Python. Для этого переходим по ссылке «<https://github.com/github/gitignore>» и скачиваем оттуда файл «Python.gitignore».

Your license is ready. Please review it below and either commit it to the main branch or to a new branch.

Py3.7 / `.gitignore` in `main` Cancel changes Commit changes...

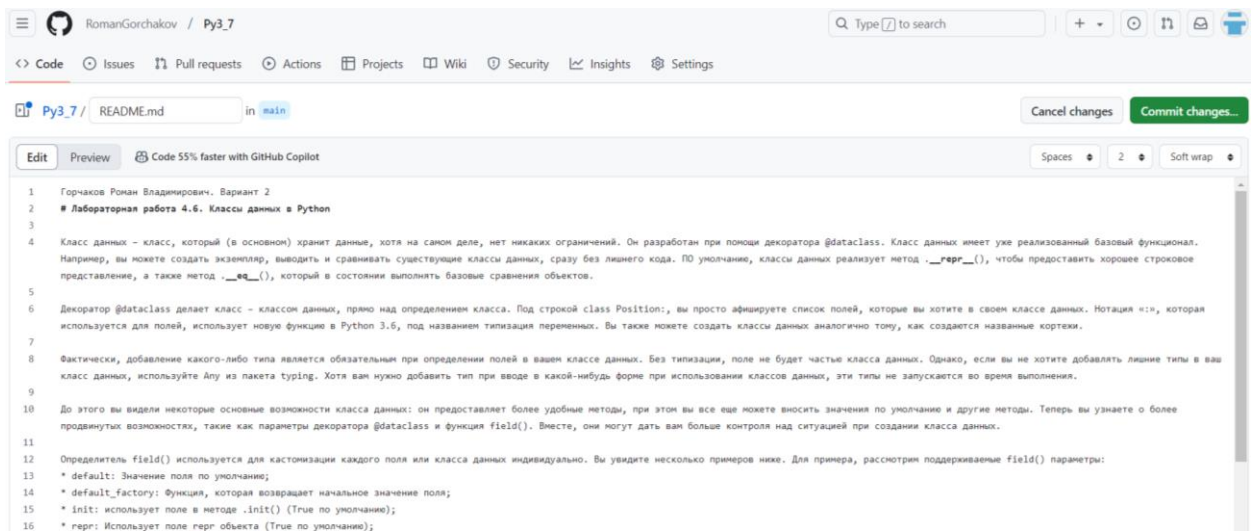
Choose .gitignore template: Python

```

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

```

3. Теперь создаём файл «README.md», где вносим ФИО и теоретический конспект лекции. Сохраняем набранный текст через кнопку «Commit changes».



4. В окне «Codespace» выбираем опцию «Create codespace on main». Откроется терминал, куда мы введём команду «git clone», чтобы клонировать свой репозиторий. После этого организуем репозиторий в соответствии с моделью ветвления Git-flow. Для этого введём в терминал команды: «git checkout -b develop» для создания ветки разработки; «git branch feature\_branch» для создания ветки функций; «git branch release/1.0.0» для создания ветки релиза; «git checkout main» и «git branch hotfix» для создания веток hotfix. Устанавливаем библиотеки isort, black и flake8 и создаём файлы .pre-commit-config.yaml и environment.yml.

```
• @RomanGorchakov →/workspaces/Py3_7 (main) $ git checkout -b develop
Switched to a new branch 'develop'
• @RomanGorchakov →/workspaces/Py3_7 (develop) $ git branch feature_branch
• @RomanGorchakov →/workspaces/Py3_7 (develop) $ git branch release/1.0.0
• @RomanGorchakov →/workspaces/Py3_7 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @RomanGorchakov →/workspaces/Py3_7 (main) $ git branch hotfix
• @RomanGorchakov →/workspaces/Py3_7 (main) $ git checkout develop
Switched to branch 'develop'
○ @RomanGorchakov →/workspaces/Py3_7 (develop) $
```

```

Collecting black
  Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.metadata (79 kB)
Collecting click>=8.0.0 (from black)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting mypy-extensions>=0.4.3 (from black)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: packaging>=22.0 in /home/codespace/.local/lib/python3.12/site-packages (from black) (24.1)
Collecting pathspec>=0.9.0 (from black)
  Downloading pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: platformdirs>=2 in /home/codespace/.local/lib/python3.12/site-packages (from black) (4.3.6)
Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl (1.8 MB)
  1.8/1.8 MB 52.8 MB/s eta 0:00:00
Downloading click-8.1.7-py3-none-any.whl (97 kB)
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading pathspec-0.12.1-py3-none-any.whl (31 kB)
Installing collected packages: pathspec, mypy-extensions, click, black
Successfully installed black-24.10.0 click-8.1.7 mypy-extensions-1.0.0 pathspec-0.12.1
● @RomanGorchakov → /workspaces/Py3_7 (develop) $ pip install flake8
Collecting flake8
  Downloading flake8-7.1.1-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting mccabe<0.8.0,>=0.7.0 (from flake8)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Collecting pycodestyle<2.13.0,>=2.12.0 (from flake8)
  Downloading pycodestyle-2.12.1-py2.py3-none-any.whl.metadata (4.5 kB)
Collecting pyflakes<3.3.0,>=3.2.0 (from flake8)
  Downloading pyflakes-3.2.0-py2.py3-none-any.whl.metadata (3.5 kB)
Downloading flake8-7.1.1-py2.py3-none-any.whl (57 kB)
Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Downloading pycodestyle-2.12.1-py2.py3-none-any.whl (31 kB)
Downloading pyflakes-3.2.0-py2.py3-none-any.whl (62 kB)
Installing collected packages: pyflakes, pycodestyle, mccabe, flake8
Successfully installed flake8-7.1.1 mccabe-0.7.0 pycodestyle-2.12.1 pyflakes-3.2.0
● @RomanGorchakov → /workspaces/Py3_7 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
● @RomanGorchakov → /workspaces/Py3_7 (develop) $ conda env export > environment.yml
○ @RomanGorchakov → /workspaces/Py3_7 (develop) $

```

5. Создаём файл «example.py», в котором нужно добавить аннотации типов для примера 1 лабораторной работы 14.

```

Command Prompt
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Примеры>python example.py
>>> add
Фамилия и инициалы? Иванов И.И.
Должность? Директор
Год поступления? 2007
>>> add
Фамилия и инициалы? Петров П.П.
Должность? Бухгалтер
Год поступления? 2010
>>> add
Фамилия и инициалы? Сидоров С.С.
Должность? Главный инженер
Год поступления? 2012
>>> list
+-----+-----+-----+-----+
| %s |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
| 1 | Иванов И.И. | Директор | 2007 |
| 2 | Петров П.П. | Бухгалтер | 2010 |
| 3 | Сидоров С.С. | Главный инженер | 2012 |
+-----+-----+-----+-----+
>>> exit
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Примеры>

```

6. Создаём файл «individual.py», в котором нужно выполнить индивидуальное задание лабораторной работы 4.5, используя классы данных, а также загрузку и сохранение данных в формат XML.

```
Command Prompt
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Индивидуальное задание>python individual.py add --db planes.db -r "Kazan" -n 3780 -t 1
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Индивидуальное задание>python individual.py add --db planes.db -r "Moscow" -n 8850 -t 9
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Индивидуальное задание>python individual.py add --db planes.db -r "Novoalexandrovsk" -n 3234 -t 62
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Индивидуальное задание>python individual.py add --db planes.db -r "Saint-Petersburg" -n 9864 -t 48
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\7\Индивидуальное задание>python individual.py add --db planes.db -r "Stavropol" -n 9120 -t 82
```

7. Выполняем коммит файлов в репозиторий Git в ветку разработки, сливаем её с веткой main и отправляем изменения на сервер GitHub.

```
Fast-forward
 .pre-commit-config.yaml      | 38 ++++++
 .python-version              | 1 +
 edu.pyoop.code-workspace     | 47 +++++++
 environment.yml              | 80 ++++++++
 pyproject.toml               | 88 ++++++++
 setup.cfg                   | 43 ++++++
 uv.lock                     | 303 ++++++
 .../individual.py"          | 165 ++++++
 ...\\320\\2404.5_\\320\\223\\320\\276\\321\\200\\321\\207\\320\\260\\320\\272\\320\\276\\320\\262\\320\\240\\320\\222.pdf" | Bin 0 -> 1205784 bytes
 "\\320\\237\\321\\200\\320\\270\\320\\274\\320\\265\\321\\200\\321\\213/example.py" | 177 ++++++
10 files changed, 942 insertions(+)
create mode 100644 .pre-commit-config.yaml
create mode 100644 .python-version
create mode 100644 edu.pyoop.code-workspace
create mode 100644 environment.yml
create mode 100644 pyproject.toml
create mode 100644 setup.cfg
create mode 100644 uv.lock
create mode 100644 "\\320\\230\\320\\275\\320\\264\\320\\270\\320\\262\\320\\270\\320\\264\\321\\203\\320\\260\\320\\273\\321\\214\\320\\275\\320\\276\\320\\265_\\320\\267\\320\\260\\320\\264\\320\\260\\320\\275\\320\\270\\320\\265\\individual.py"
create mode 100644 "\\320\\236\\321\\202\\321\\207\\321\\221\\321\\202\\320\\233\\320\\2404.5_\\320\\223\\320\\276\\321\\200\\321\\207\\320\\260\\320\\272\\320\\276\\320\\262\\320\\240\\320\\222.pdf"
create mode 100644 "\\320\\237\\321\\200\\320\\270\\320\\274\\320\\265\\321\\200\\321\\213/example.py"
@RomanGorchakov -> /workspaces/Py3_7 (main) $ git push -u
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (15/15), 1.09 MiB | 3.67 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RomanGorchakov/Py3_7
 503cd8a..c0a5f14 main -> main
branch 'main' set up to track 'origin/main'.
@RomanGorchakov -> /workspaces/Py3_7 (main) $
```

Py3\_7 Public

main 1 Branch 0 Tags

Go to file Add file Code

RomanGorchakov Data classes c0a5f14 · now 4 Commits

Индивидуальное задание	Data classes	now
Отчёт	Data classes	now
Примеры	Data classes	now
.gitignore	Create .gitignore	5 minutes ago
.pre-commit-config.yaml	Data classes	now
.python-version	Data classes	now
LICENSE	Create LICENSE	6 minutes ago
README.md	Create README.md	5 minutes ago
edu.pyoop.code-workspace	Data classes	now
environment.yml	Data classes	now
pyproject.toml	Data classes	now
setup.cfg	Data classes	now
uv.lock	Data classes	now

README MIT license

Горчаков Роман Владимирович. Вариант 2

## Лабораторная работа 4.6. Классы данных в Python

About: No description, website, or topics provided. Readme MIT license Activity 0 stars 1 watching 0 forks

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

Languages: Python 100.0%

Suggested workflows: Based on your tech stack. Pylint (Lint a Python application with pylint), Python application (Create and test a Python application).

## Контрольные вопросы

### 1. Как создать класс данных в языке Python?

Чтобы создать класс данных в Python, можно воспользоваться декоратором `@dataclass`.

```
from dataclasses import dataclass
```

```
@dataclass
class DataClassCard:
    rank: str
    suit: str
```

### 2. Какие методы по умолчанию реализует класс данных?

Класс данных имеет уже реализованный базовый функционал. Например, вы можете создать экземпляр, выводить и сравнивать существующие классы данных, сразу без лишнего кода. По умолчанию, классы данных реализует метод `__repr__()`

, чтобы предоставить хорошее строковое представление, а также метод `__eq__()`, который в состоянии выполнять базовые сравнения объектов.

Класс данных – обычный класс Python. Единственное, что его отличает, это то, он содержит базовые методы модели данных, такие как `__init__()`, `__repr__()` и `__eq__()`.

### 3. Как создать неизменяемый класс данных?

Чтобы сделать класс данных неизменяемым, установите `frozen=True` при создании.

```
from dataclasses import dataclass
```

```
@dataclass (frozen=True)
```

```
class Position:
```

```
    name: str
```

```
    lon: float = 0.0
```

```
    lat: float = 0.0
```