

Лабораторная работа 2.25


Тема: Управление процессами в Python

Цель работы: приобретение навыков написания многозадачных приложений на языке программирования Python версии 3.x.

Ссылка на GitHub: https://github.com/RomanGorchakov/Py3_12

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_12 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	Description
Apache License 2.0	A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.
GNU General Public License v3.0	
MIT License	<div> <div> Permissions <ul style="list-style-type: none"> ✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use </div> <div> Limitations <ul style="list-style-type: none"> ✗ Liability ✗ Warranty </div> <div> Conditions <ul style="list-style-type: none"> ① License and copyright notice </div> </div> <p>This is not legal advice. Learn more about repository licenses.</p>
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	

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 ①

Full name ①

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».

RomanGorchakov / Py3_12

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

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

Py3_12 / .gitignore in main

Cancel changes Commit changes...

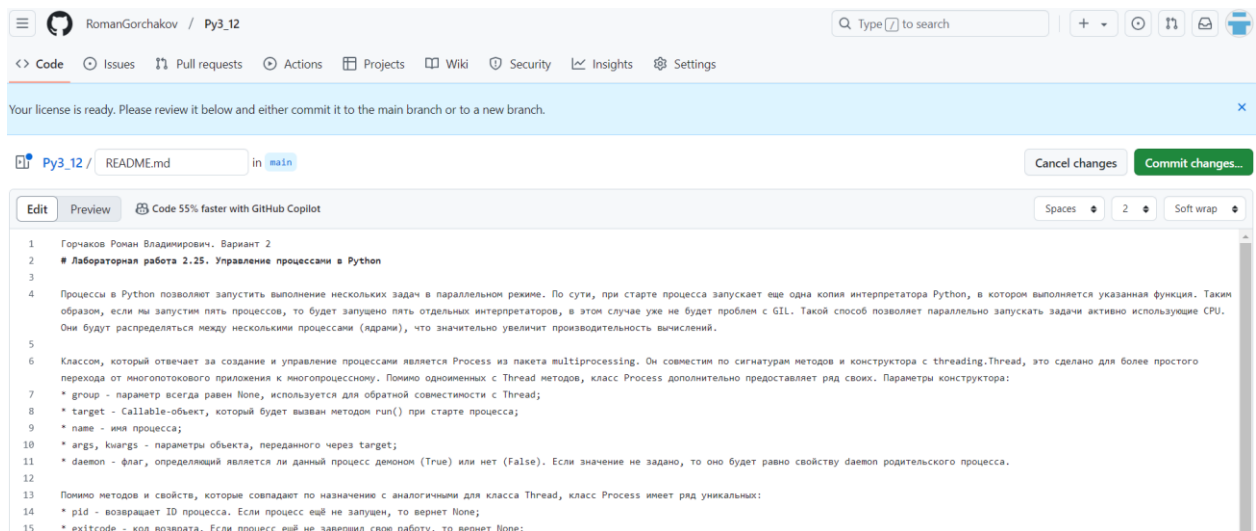
Edit Preview Choose .gitignore template: Python Code 55% faster with GitHub Copilot Spaces 2 No wrap

```

1 # Byte-compiled / optimized / DLL files
2 __pycache__/
3 *.py[od]
4 *.pyc
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 соответствие 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.

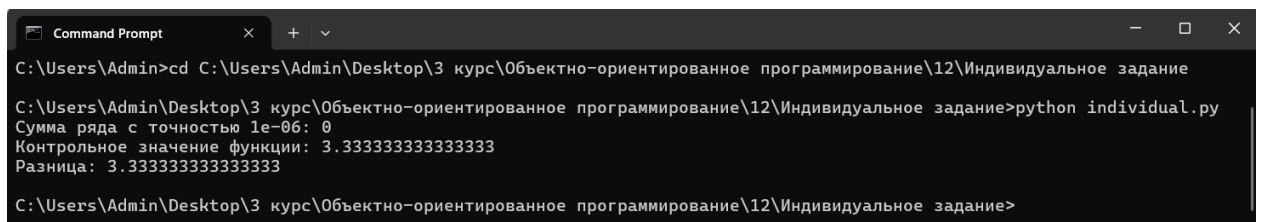
```
• @RomanGorchakov → /workspaces/Py3_12 (main) $ git checkout -b develop
Switched to a new branch 'develop'
• @RomanGorchakov → /workspaces/Py3_12 (develop) $ git branch feature_branch
• @RomanGorchakov → /workspaces/Py3_12 (develop) $ git branch release/1.0.0
• @RomanGorchakov → /workspaces/Py3_12 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @RomanGorchakov → /workspaces/Py3_12 (main) $ git branch hotfix
• @RomanGorchakov → /workspaces/Py3_12 (main) $ git checkout develop
Switched to branch 'develop'
○ @RomanGorchakov → /workspaces/Py3_12 (develop) $
```

```

• @RomanGorchakov → /workspaces/Py3_12 (develop) $ pip install black
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-8.1.8-py3-none-any.whl.metadata (2.3 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) (24.2)
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 45.5 MB/s eta 0:00:00
Downloading click-8.1.8-py3-none-any.whl (98 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-8.1.8 mypy_extensions-1.0.0 pathspec-0.12.1
• @RomanGorchakov → /workspaces/Py3_12 (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_12 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
• @RomanGorchakov → /workspaces/Py3_12 (develop) $ conda env export > environment.yml

```

5. Создаём файл «individual.py», в котором нужно реализовать вычисление значений в двух функций в отдельных процессах для индивидуального задания лабораторной работы 2.23.



```

C:\Users\Admin>cd C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\12\Индивидуальное задание
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\12\Индивидуальное задание>python individual.py
Сумма ряда с точностью 1e-06: 0
Контрольное значение функции: 3.333333333333333
Разница: 3.333333333333333
C:\Users\Admin\Desktop\3 курс\Объектно-ориентированное программирование\12\Индивидуальное задание>

```

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

```

@RomanGorchakov →/workspaces/Py3_12 (main) $ git merge develop
Updating 38aeaae..6b8125c
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            | 45 ++++++
 ...320\2402.23_320\223\320\276\321\200\321\207\320\260\320\272\320\276\320\262\320\240\320\222.pdf" | Bin 0 -> 896620 bytes
9 files changed, 645 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\2402.23_320\223\320\276\321\200\321\207\320\260\320\272\320\276\320\262\320\240\320\222.pdf"
@RomanGorchakov →/workspaces/Py3_12 (main) $ git push -u
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 2 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (13/13), 841.52 KiB | 22.14 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RomanGorchakov/Py3_12
 38aeaae..6b8125c main -> main
branch 'main' set up to track 'origin/main'.
@RomanGorchakov →/workspaces/Py3_12 (main) $

```

Py3_12 Public

Pin

Unwatch 1

Fork 0

Star 0

main 1 Branch 0 Tags

Go to file

Add file

Code

RomanGorchakov Multiprocessing 6b8125c · now 4 Commits

Индивидуальное задание	Multiprocessing	now
Отчёт	Multiprocessing	now
.gitignore	Create .gitignore	20 minutes ago
.pre-commit-config.yaml	Multiprocessing	now
.python-version	Multiprocessing	now
LICENSE	Create LICENSE	20 minutes ago
README.md	Create README.md	12 minutes ago
edu.pyoop.code-workspace	Multiprocessing	now
environment.yml	Multiprocessing	now
pyproject.toml	Multiprocessing	now
setup.cfg	Multiprocessing	now
uv.lock	Multiprocessing	now

README MIT license

Горчаков Роман Владимирович. Вариант 2

Лабораторная работа 2.25. Управление процессами в Python

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

Python Package using Anaconda

Configure

Create and test a Python package on multiple Python versions using Anaconda for package management.

Контрольные вопросы

1. Как создаются и завершаются процессы в Python?

Создание процессов в Python происходит с помощью класса `Process` из пакета `multiprocessing`. Для запуска процесса используется метод `start()`.

За ожидание завершения работы процесса отвечает метод `join()`. При выводе этого метода выполнение программы будет остановлено до тех пор, пока соответствующий процесс не завершит работу. Параметр `timeout` отвечает за время ожидания завершения работы процесса, если указанное время прошло, а процесс ещё не завершился, то ожидание будет прервано и выполнение программы продолжится дальше.

2. В чем особенность создания классов-наследников от `Process`?

В классе-наследнике от `Process` необходимо переопределить метод `run()` для того, чтобы он (класс) соответствовал протоколу работы с процессами.

3. Как выполнить принудительное завершение процесса?

Принудительное завершение работы процессов осуществляется с помощью методов `terminate()` или `kill()` класса `Process`. В Unix отправляется команда `SIGTERM`, в Windows используется функция `TerminateProcess`.

4. Что такое процессы-демоны? Как запустить процесс-демон?

Процессы демоны по своим свойствам похожи на потоки-демоны, их суть заключается в том, что они завершают свою работу, если завершился родительский процесс.

Указание на то, что процесс является демоном должно быть сделано до его запуска (до вызова метода `start()`). Для демонического процесса запрещено самостоятельно создавать дочерние процессы. Эти процессы не являются демонами (сервисами) в понимании Unix, единственное их свойство – завершение работы вместе с родительским процессом.

Указать на то, что процесс является демоном можно при создании экземпляра класса через аргумент `daemon`, либо после создания через свойство `daemon`.