


Лабораторная работа 2

Тема: Структура программы и управление потоком выполнения

Ссылка на GitHub: https://github.com/RomanGorchakov/Rust_2

Порядок выполнения работы

1. Создаём аккаунт в GitHub. Затем 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 *

Rust_2

✓ Rust_2 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	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	<ul style="list-style-type: none"> ✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use 	<ul style="list-style-type: none"> ✗ Liability ✗ Warranty 	<ul style="list-style-type: none"> ① License and copyright notice
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			

This is not legal advice. [Learn more about repository licenses.](#)

MIT License

Copyright (c) Year Full name

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:

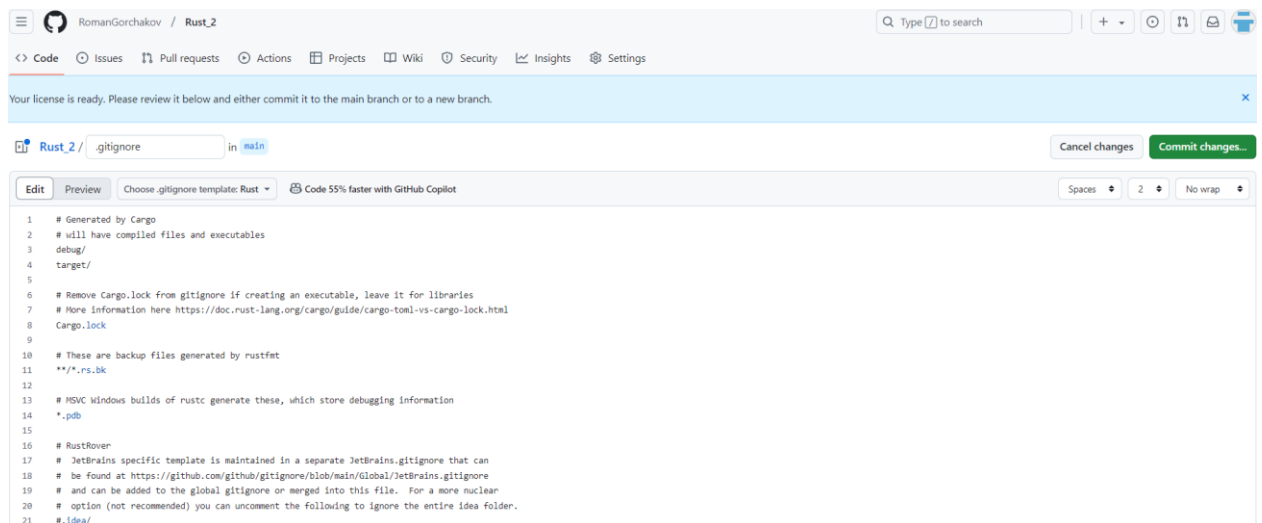
To adopt MIT License, enter your details. You'll have a chance to review before committing a `LICENSE` file to a new branch or the root of your project.

Year ①

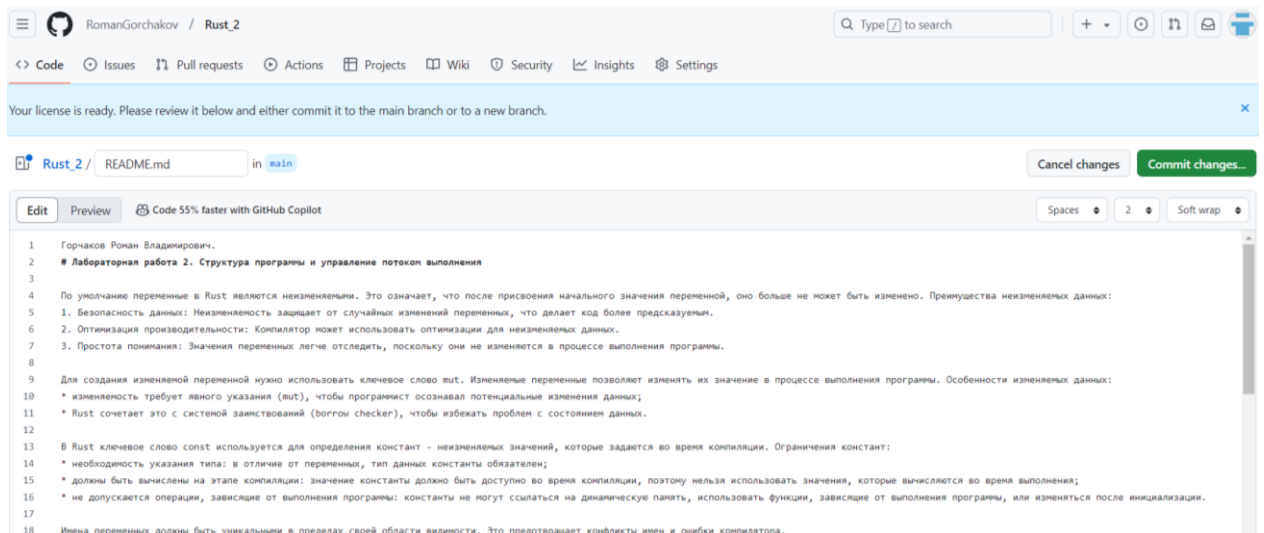
Full name ①

[Review and submit](#)

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



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/Rust_2 (main) $ git checkout -b develop
Switched to a new branch 'develop'
• @RomanGorchakov →/workspaces/Rust_2 (develop) $ git branch feature_branch
• @RomanGorchakov →/workspaces/Rust_2 (develop) $ git branch release/1.0.0
• @RomanGorchakov →/workspaces/Rust_2 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @RomanGorchakov →/workspaces/Rust_2 (main) $ git branch hotfix
• @RomanGorchakov →/workspaces/Rust_2 (main) $ git checkout develop
Switched to branch 'develop'
○ @RomanGorchakov →/workspaces/Rust_2 (develop) $
```

```

• @RomanGorchakov →/workspaces/Rust_2 (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-8.1.7-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) (24.2)
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 64.4 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/Rust_2 (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/Rust_2 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
• @RomanGorchakov →/workspaces/Rust_2 (develop) $ conda env export > environment.yml

```

5. Создаём файл «example1.rs», который вычисляет площадь круга.

```

src > main.rs > main
1 fn main() {
2     let radius: f64 = 3.0;
3     let area = std::f64::consts::PI * radius.powi(2);
4     println!("Площадь круга: {}", area);
5 }

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - src
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\2\second_rust_project\src> rustc main.rs
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\2\second_rust_project\src> ./main.exe
Площадь круга: 28.274333882308138
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\2\second_rust_project\src>

```

6. Создаём файл «example2.rs», в котором нужно произвести итерацию с дробным шагом.

```
main.rs U X
src > main.rs > main
1 fn main() {
2     for i in (0..10).map(|x| x as f64 * 0.1) {
3         println!("{:.1}", i);
4     }
5 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS powershell - src

```
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> rustc main.rs
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> .\main.exe
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src>
```

7. Создаём файл «example3.rs», в котором вывести на экран результат работы операторов отношений.

```
main.rs U X
src > main.rs > main
1 fn main() {
2     let a = 7;
3     let b = 10;
4
5     println!("a == b: {}", a == b);
6     println!("a != b: {}", a != b);
7     println!("a > b: {}", a > b);
8     println!("a < b: {}", a < b);
9     println!("a >= b: {}", a >= b);
10    println!("a <= b: {}", a <= b);
11 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS powershell - src

```
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> rustc main.rs
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> .\main.exe
a == b: false
a != b: true
a > b: false
a < b: true
a >= b: false
a <= b: true
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src>
```

8. Создаём файл «example4.rs», в котором нужно вывести на экран результат работы операторов побитовых отношений.

```
src > main.rs U X
```

```
1 fn main() {  
2     let a = 0b1100;  
3     let b = 0b1010;  
4  
5     println!("а и b: {:04b}", a & b);  
6     println!("а ИЛИ b: {:04b}", a | b);  
7     println!("а XOR b: {:04b}", a ^ b);  
8     println!("HE a: {:04b}", !a);  
9     println!("a << 1: {:04b}", a << 1);  
10    println!("a >> 2: {:04b}", a >> 2);  
11 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - src + - [ ] [X] ... ^ >
```

```
● PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project> cd src  
● PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> rustc main.rs  
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> .\main.exe  
  
а и b: 1000  
а ИЛИ b: 1110  
а XOR b: 0110  
HE a: 111111111111111111111111111111110011  
а << 1: 11000  
а >> 2: 0011  
PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> 
```

9. Создаём файл «task.rs», в котором нужно проверить введённый пользователем пароль на наличие определённых символов.

```
src > main.rs > main
1 use std::io;
2
3 fn main() {
4     println!("Введите пароль:");
5     let mut password = String::new();
6
7     io::stdin()
8         .read_line(&mut password)
9         .expect("Не удалось прочитать строку");
10
11     let password = password.trim();
12
13     if password.chars().any(|c| c.is_numeric()) &&
14        password.chars().any(|c| c.is_alphabetic()) {
15         println!("Пароль надёжный!");
16     } else {
17         println!("Пароль должен содержать буквы и цифры!");
18     }
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell - src + ▢ 🗑️ ⋮ ^ ×

- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> rustc main.rs
- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> .\main.exe

Введите пароль:
15ZPBQF2h
Пароль надёжный!

PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\2\second_rust_project\src> ▢

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

```
Your branch is up to date with 'origin/main'.
● @RomanGorchakov →/workspaces/Rust_2 (main) $ git merge develop
Updating 8d62b16..98b964d
Fast-forward
 .pre-commit-config.yaml | 38 ++++++
 Cargo.toml               | 6 +++
 environment.yml          | 80 ++++++
 "...33\320\2401_\320\223\320\276\321\200\321\207\320\260\320\272\320\276\320\262\320\240\320\222.pdf" | 18 ++++++
 Bin 0 -> 1232748 bytes
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\example1.rs" | 5 +++
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\example2.rs" | 5 +++
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\example3.rs" | 11 +++++
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213\example4.rs" | 12 +++++
 9 files changed, 175 insertions(+)
 create mode 100644 .pre-commit-config.yaml
 create mode 100644 Cargo.toml
 create mode 100644 environment.yml
 create mode 100644 "...320\227\320\260\320\264\320\260\320\275\320\270\320\265\320\265\320\272\320\276\320\262\320\240\320\222.pdf"
 create mode 100644 "...320\236\321\202\321\207\321\221\321\202\320\233\320\2401_\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\example1.rs"
 create mode 100644 "...320\237\321\200\320\270\320\274\320\265\321\200\321\213\example2.rs"
 create mode 100644 "...320\237\321\200\320\270\320\274\320\265\321\200\321\213\example3.rs"
 create mode 100644 "...320\237\321\200\320\270\320\274\320\265\321\200\321\213\example4.rs"
● @RomanGorchakov →/workspaces/Rust_2 (main) $ git push -u
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 2 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (14/14), 1.12 MiB | 4.74 MiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RomanGorchakov/Rust_2
 8d62b16..98b964d main -> main
branch 'main' set up to track 'origin/main'.
○ @RomanGorchakov →/workspaces/Rust_2 (main) $
```

Rust_2Public

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to fileAdd fileCode

RomanGorchakovStructure98b964d · 5 minutes ago4 Commits

Задание	Structure	5 minutes ago
Отчёт	Structure	5 minutes ago
Примеры	Structure	5 minutes ago
.gitignore	Create .gitignore	2 days ago
.pre-commit-config.yaml	Structure	5 minutes ago
Cargo.toml	Structure	5 minutes ago
LICENSE	Create LICENSE	2 days ago
README.md	Create README.md	2 days ago
environment.yml	Structure	5 minutes ago

READMEMIT license

Горчаков Роман Владимирович.

Лабораторная работа 2. Структура программы и управление потоком выполнения

AboutNo description, website, or topics provided.

ReleasesNo releases published

PackagesNo packages published

LanguagesRust 100.0%

Suggested workflowsBased on your tech stack