

# Лабораторная работа 1

Тема: Введение в Rust

Цель работы: приобретение навыков по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x

Ссылка на GitHub: [https://github.com/RomanGorchakov/Rust\\_1](https://github.com/RomanGorchakov/Rust_1)


## Порядок выполнения работы

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

### 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 *	Repository name *
 RomanGorchakov ▾	/ <input type="text" value="Test"/>
✔ Test is available.	

Great repository names are short and memorable. Need inspiration? How about [automatic-octo-chainsaw](#) ?

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

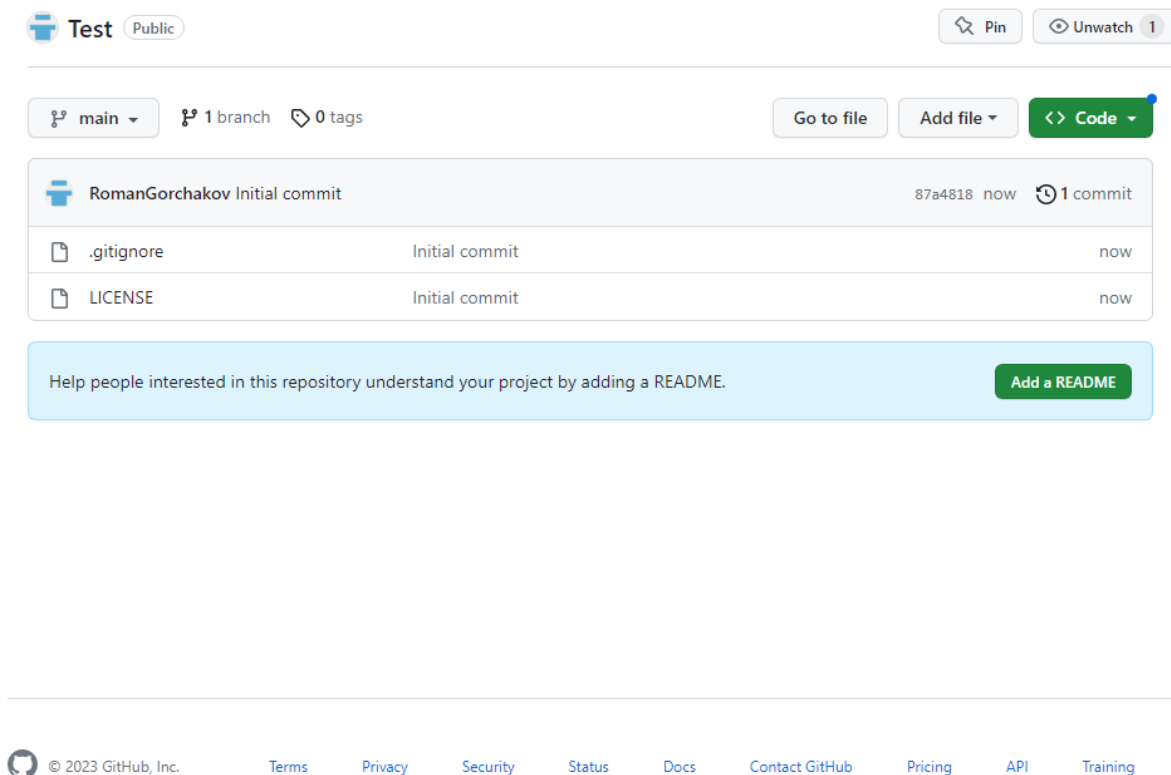
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a 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



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`».

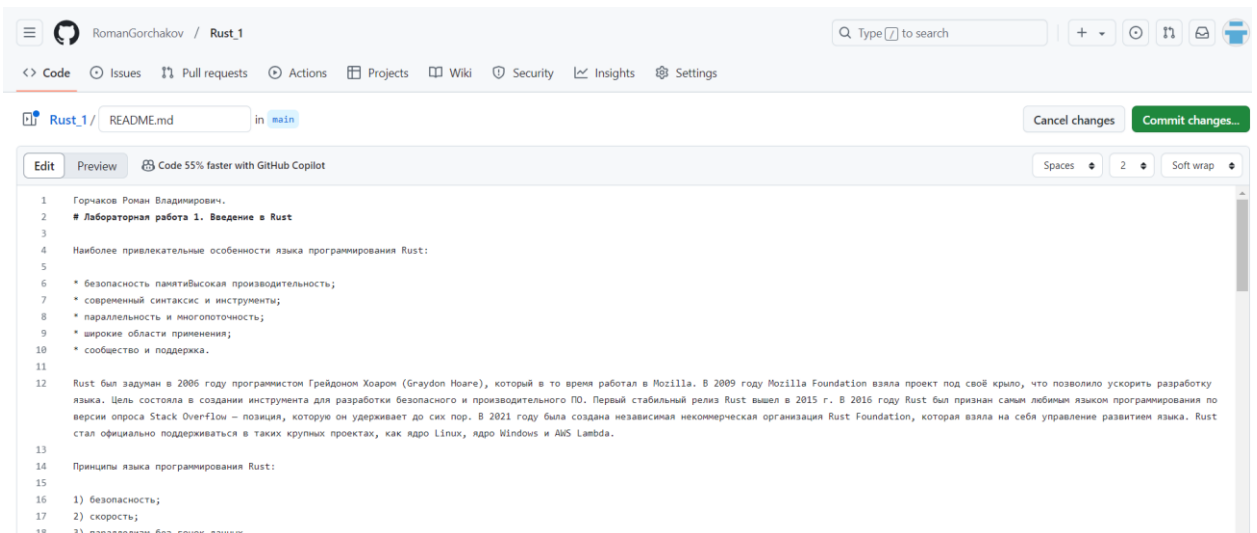
 <code>Racket.gitignore</code>	Update Racket.gitignore	2 years ago
 <code>Rails.gitignore</code>	Ignore Rails .env according recommendations	3 years ago
 <code>Raku.gitignore</code>	Changes the name of Perl 6 to Raku (#3312)	4 years ago
 <code>ReScript.gitignore</code>	Update ReScript.gitignore	6 months ago
 <code>RhodesRhomobile.gitignore</code>	Add Rhodes mobile application framework gitignore	14 years ago
 <code>Ruby.gitignore</code>	Ruby: ignore RuboCop remote inherited config files (#3197)	5 years ago
 <a href="#">Rust.gitignore</a>	Add RustRover entry for Rust.gitignore	7 months ago
 <code>SCons.gitignore</code>	Update SCons.gitignore (#3001)	5 years ago
 <code>Sass.gitignore</code>	Add all sourcemap file formats	6 years ago
 <code>Scala.gitignore</code>	add scala jvm crash files	3 years ago
 <code>Scheme.gitignore</code>	Add Scheme.gitignore	9 years ago
 <code>Scrivener.gitignore</code>	Added new checksums and allows for relative directories (#3...	5 years ago
 <code>Sdcc.gitignore</code>	ensure single trailing newline	11 years ago
 <code>SeamGen.gitignore</code>	Update SeamGen.gitignore	12 years ago
 <code>SketchUp.gitignore</code>	Add gitignore for sketchup backup files.	13 years ago
 <code>Smalltalk.gitignore</code>	Merge pull request #2027 from rejurime/dolphin-smalltalk	6 years ago

```

1  # Generated by Cargo
2  # will have compiled files and executables
3  debug/
4  target/
5
6  # Remove Cargo.lock from gitignore if creating an executable, leave it for libraries
7  # More information here https://doc.rust-lang.org/cargo/guide/cargo-toml-vs-cargo-lock.html
8  Cargo.lock
9
10 # These are backup files generated by rustfmt
11 **/*.rs.bk
12
13 # MSVC Windows builds of rustc generate these, which store debugging information
14 *.pdb
15
16 # RustRover
17 # JetBrains specific template is maintained in a separate JetBrains.gitignore that can
18 # be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
19 # and can be added to the global gitignore or merged into this file. For a more nuclear
20 # option (not recommended) you can uncomment the following to ignore the entire idea folder.
21 #.idea/

```

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

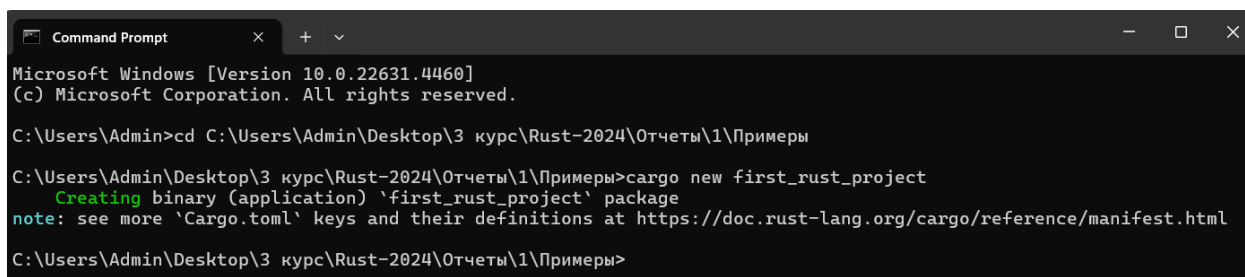
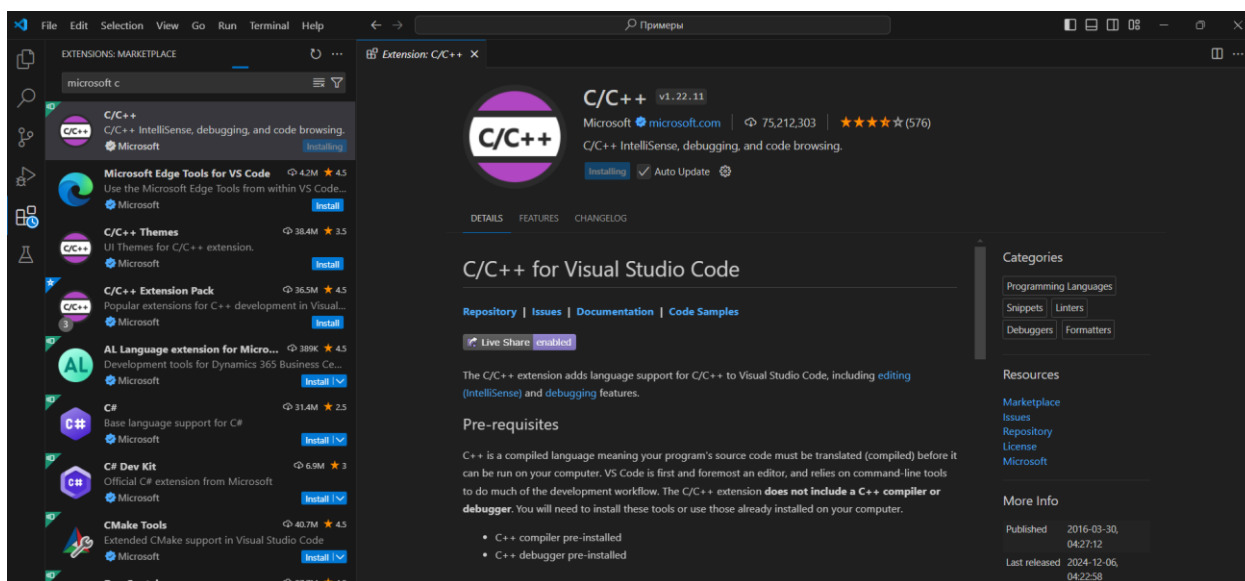
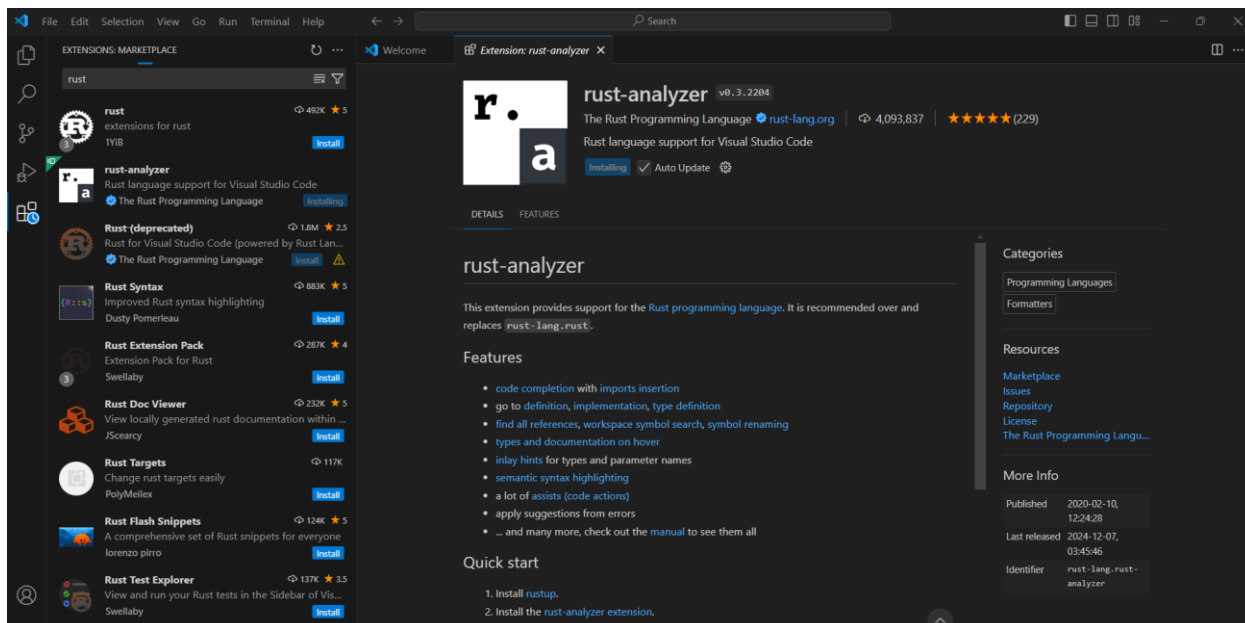
develop» для создания ветки разработки; «git branch feature\_branch» для создания ветки функций; «git branch release/1.0.0» для создания ветки релиза; «git checkout main» и «git branch hotfix» для создания веток hotfix. Устанавливаем библиотеки isort, black и flake8 и создаём файлы .pre-commit-config.yaml и environment.yml.

```
• @RomanGorchakov →/workspaces/Rust_1 (main) $ git checkout -b develop
Switched to a new branch 'develop'
• @RomanGorchakov →/workspaces/Rust_1 (develop) $ git branch feature_branch
• @RomanGorchakov →/workspaces/Rust_1 (develop) $ git branch release/1.0.0
• @RomanGorchakov →/workspaces/Rust_1 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @RomanGorchakov →/workspaces/Rust_1 (main) $ git branch hotfix
• @RomanGorchakov →/workspaces/Rust_1 (main) $ git checkout develop
Switched to branch 'develop'
○ @RomanGorchakov →/workspaces/Rust_1 (develop) $
```

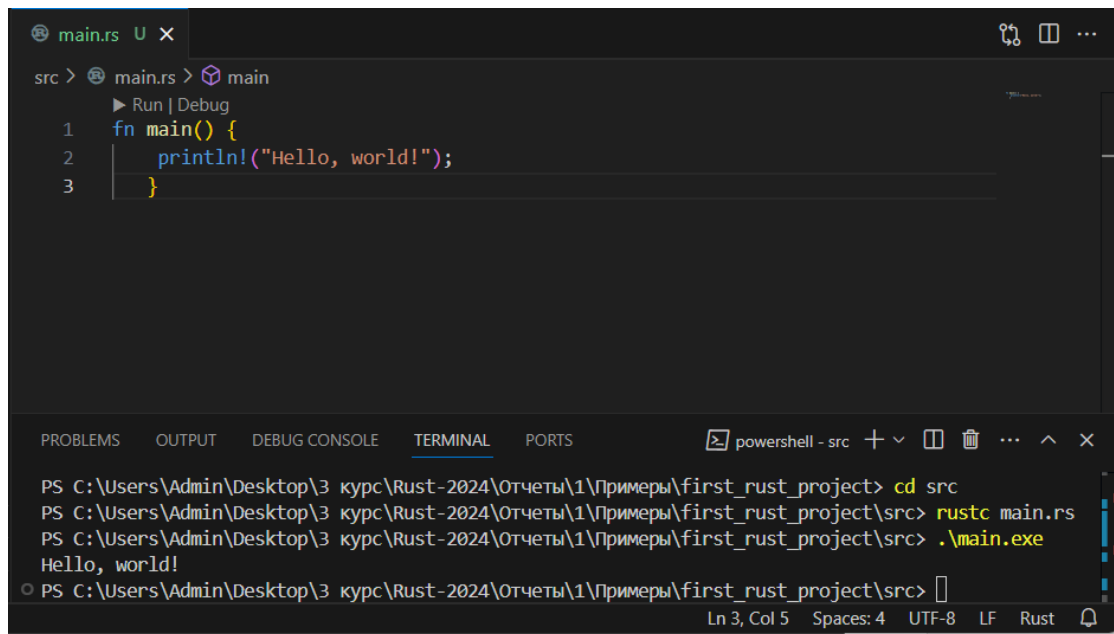
```
Collecting black
  Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.metadata (79 kB)
Collecting click>=8.0.0 (from black)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting mypy_extensions>=0.4.3 (from black)
  Downloading 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 60.2 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_1 (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_1 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
• @RomanGorchakov →/workspaces/Rust_1 (develop) $ conda env export > environment.yml
○ @RomanGorchakov →/workspaces/Rust_1 (develop) $
```

5. Переходим по ссылке <https://www.rust-lang.org/tools/install> на официальный сайт Rust и устанавливаем Rust. После этого устанавливаем Microsoft

Visual Studio Code, затем устанавливаем расширения rust-analyzer и Microsoft C/C++. Открываем командную строку и вводим в неё команду `cargo new first_rust_project`.



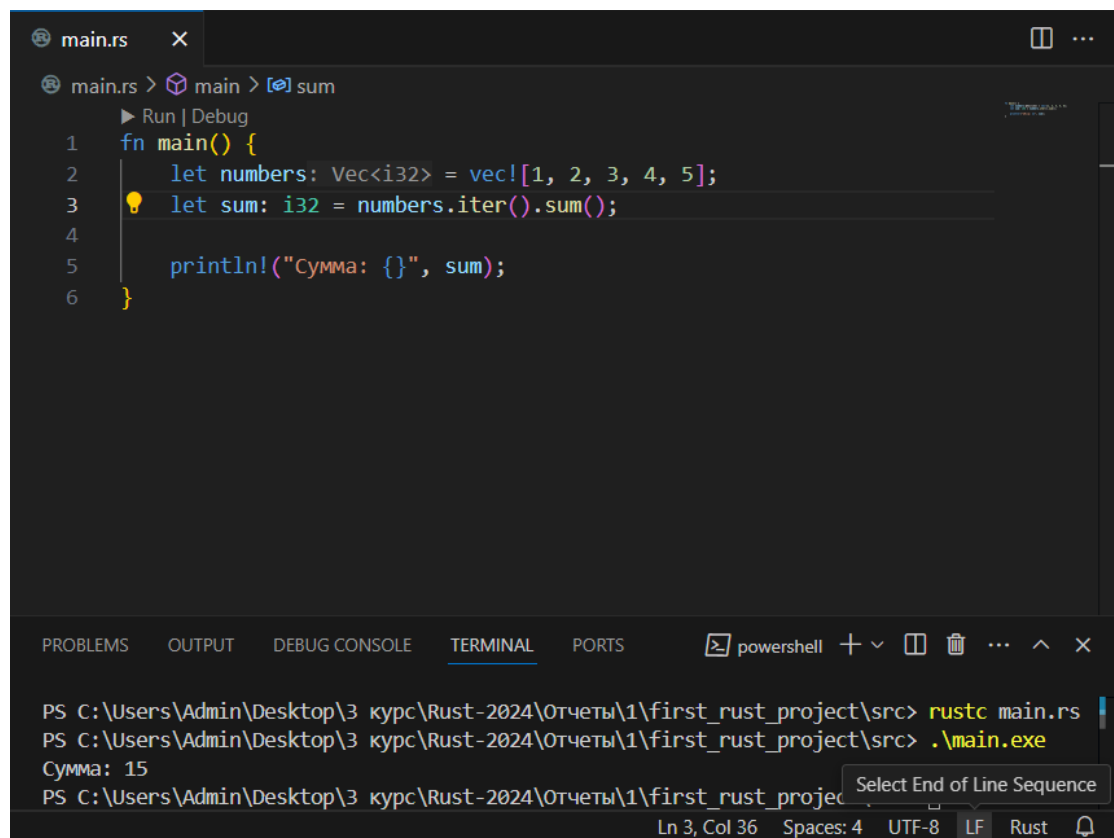
6. Открываем Microsoft Visual Studio Code из созданной папки. Создаём файл «example.rs», который выводит надпись «Hello, world!».



```
main.rs U x
src > main.rs > main
  ▶ Run | Debug
1  fn main() {
2      println!("Hello, world!");
3  }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell - src
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\Примеры\first_rust_project> cd src
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\Примеры\first_rust_project\src> rustc main.rs
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\Примеры\first_rust_project\src> .\main.exe
Hello, world!
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\Примеры\first_rust_project\src>
Ln 3, Col 5 Spaces: 4 UTF-8 LF Rust
```

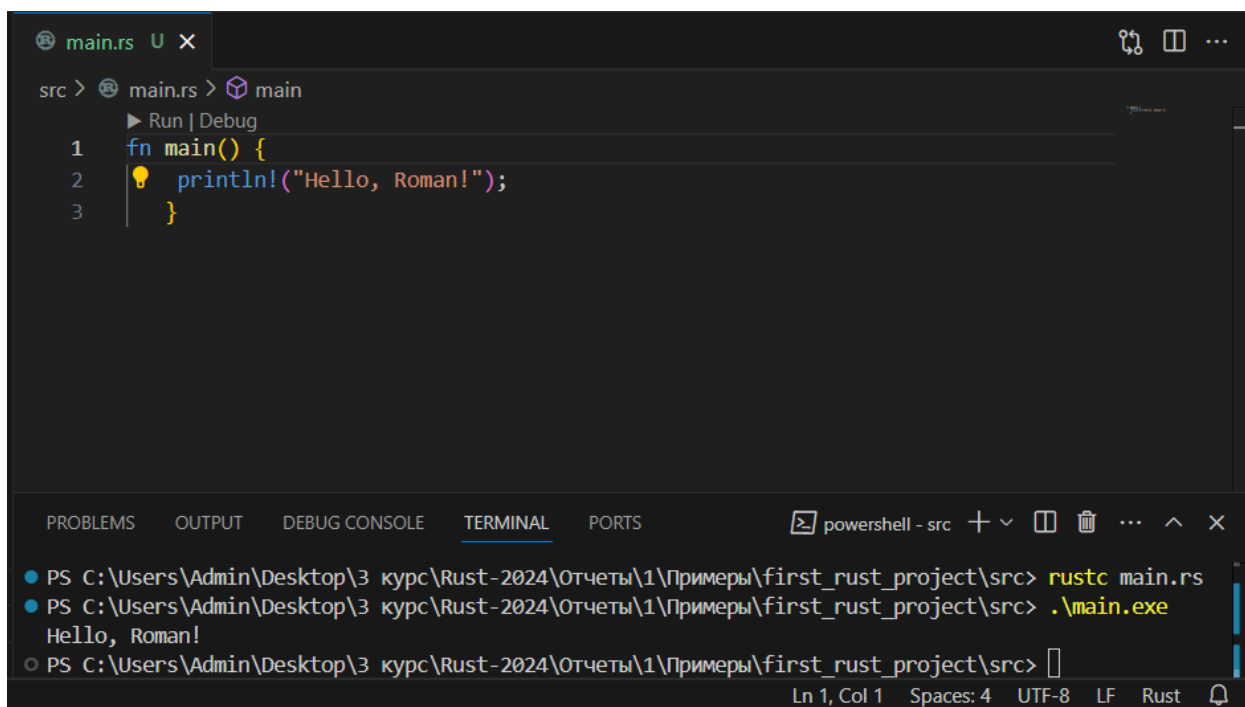
7. Создаём файл «example2.py», в котором нужно вывести на экран сумму чисел в массиве.



```
main.rs x
main.rs > main > sum
  ▶ Run | Debug
1  fn main() {
2      let numbers: Vec<i32> = vec![1, 2, 3, 4, 5];
3      let sum: i32 = numbers.iter().sum();
4
5      println!("Сумма: {}", sum);
6  }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + v
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\first_rust_project\src> rustc main.rs
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\first_rust_project\src> .\main.exe
Сумма: 15
PS C:\Users\Admin\Desktop\3 курс\Rust-2024\Отчеты\1\first_rust_project\src>
Ln 3, Col 36 Spaces: 4 UTF-8 LF Rust
```

8. Создаём файл «task1.py», в котором нужно вывести на экран имя пользователя.



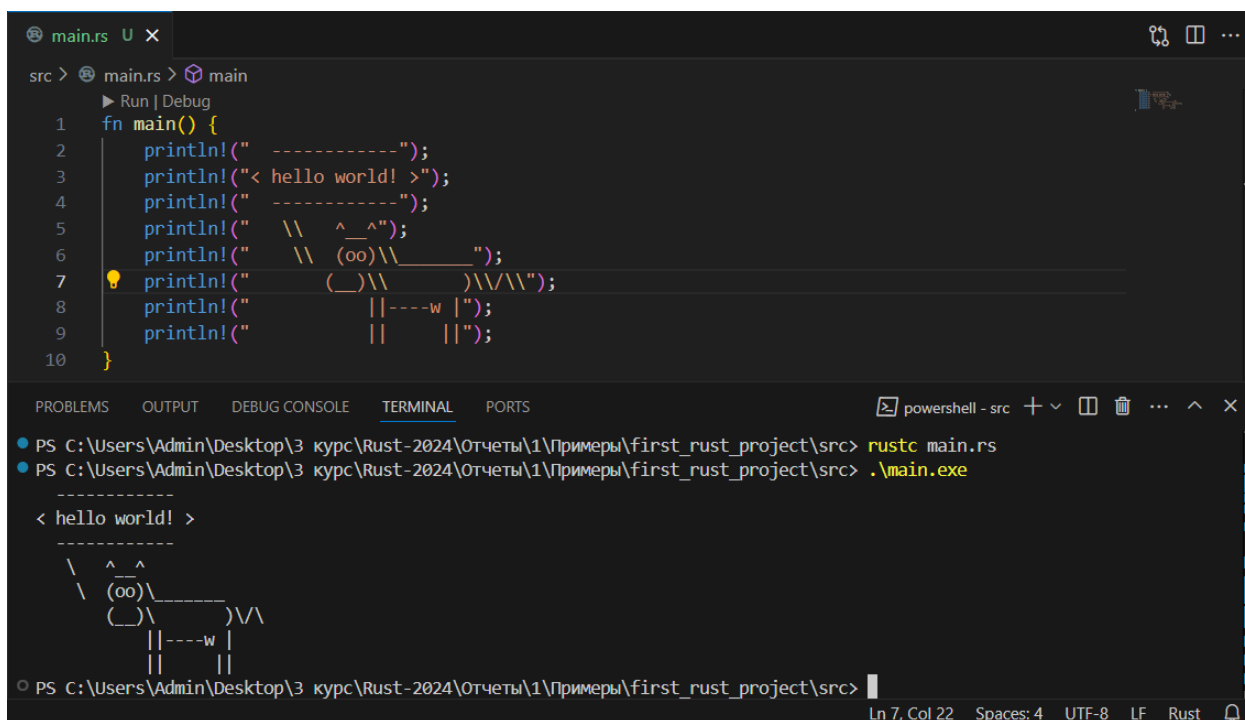
```
main.rs U x
src > main.rs > main
  ▶ Run | Debug
1  fn main() {
2      println!("Hello, Roman!");
3  }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS powershell - src + - ▢ 🗑 ... ^ x

- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src> rustc main.rs
- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src> .\main.exe  
Hello, Roman!
- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src>

Ln 1, Col 1 Spaces: 4 UTF-8 LF Rust 🔔

9. Создаём файл «task2.rs», в котором нужно создать графическое изображение с помощью нескольких макросов println!



```
main.rs U x
src > main.rs > main
  ▶ Run | Debug
1  fn main() {
2      println!(" -----");
3      println!("< hello world! >");
4      println!(" -----");
5      println!("      \  ^__^");
6      println!("      \  (oo)\_____");
7      println!("      (__)\       )\/\");
8      println!("      ||----w |");
9      println!("      ||     ||");
10 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS powershell - src + - ▢ 🗑 ... ^ x

- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src> rustc main.rs
- PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src> .\main.exe

```
< hello world! >
-----
 \  ^__^
 \  (oo)\_____
  (__)\       )\/\
    ||----w |
    ||     ||
```

PS C:\Users\Admin\Desktop\3 кypc\Rust-2024\Отчеты\1\Примеры\first\_rust\_project\src>

Ln 7, Col 22 Spaces: 4 UTF-8 LF Rust 🔔

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

```
@RomanGorchakov →/workspaces/Rust_1 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
@RomanGorchakov →/workspaces/Rust_1 (main) $ git merge develop
Updating 336d63c..3c9c91a
Fast-forward
 .pre-commit-config.yaml      | 38 +++++
 Cargo.toml                   | 6 +++
 environment.yml               | 80 +++++
 .../320\227\320\260\320\264\320\260\320\275\320\270\320\265 1/task1.rs" | 3 ++
 .../320\227\320\260\320\264\320\260\320\275\320\270\320\265 2/task2.rs" | 10 +++++
 .../320\2404.3 \320\223\320\276\321\200\321\207\320\260\320\272\320\276\320\262\320\240\320\222.pdf" | Bin 0 -> 922941 bytes
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213/example1.rs" | 3 ++
 "\320\237\321\200\320\270\320\274\320\265\321\200\321\213/example2.rs" | 6 +++
 8 files changed, 146 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\321\217\320\227\320\260\320\264\320\260\320\275\320\270\320\265 1/task1.rs"
 create mode 100644 ".../320\227\320\260\320\264\320\260\320\275\320\270\321\217\320\227\320\260\320\264\320\260\320\275\320\270\320\265 2/task2.rs"
 create mode 100644 ".../320\236\321\202\321\207\321\221\321\202\320\233\320\2404.3 \320\223\320\260\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"
@RomanGorchakov →/workspaces/Rust_1 (main) $ git push -u
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 851.80 KiB | 20.78 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RomanGorchakov/Rust_1
 336d63c..3c9c91a  main -> main
branch 'main' set up to track 'origin/main'.
@RomanGorchakov →/workspaces/Rust_1 (main) $
```

Rust\_1Public

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to fileAdd fileCode

About

RomanGorchakovRust3c9c91a · now5 Commits

Задания	Rust	now
Отчёт	Rust	now
Примеры	Rust	now
.gitignore	Update .gitignore	3 days ago
.pre-commit-config.yaml	Rust	now
Cargo.toml	Rust	now
LICENSE	Create LICENSE	3 days ago
README.md	Create README.md	2 days ago
environment.yml	Rust	now

READMEMIT license

Горчаков Роман Владимирович.

Лабораторная работа 1. Введение в Rust

No description, website, or topics provided.

ReadmeMIT licenseActivity0 stars1 watching0 forks

ReleasesNo releases publishedCreate a new release

PackagesNo packages publishedPublish your first package

LanguagesRust 100.0%

Suggested workflowsBased on your tech stack