

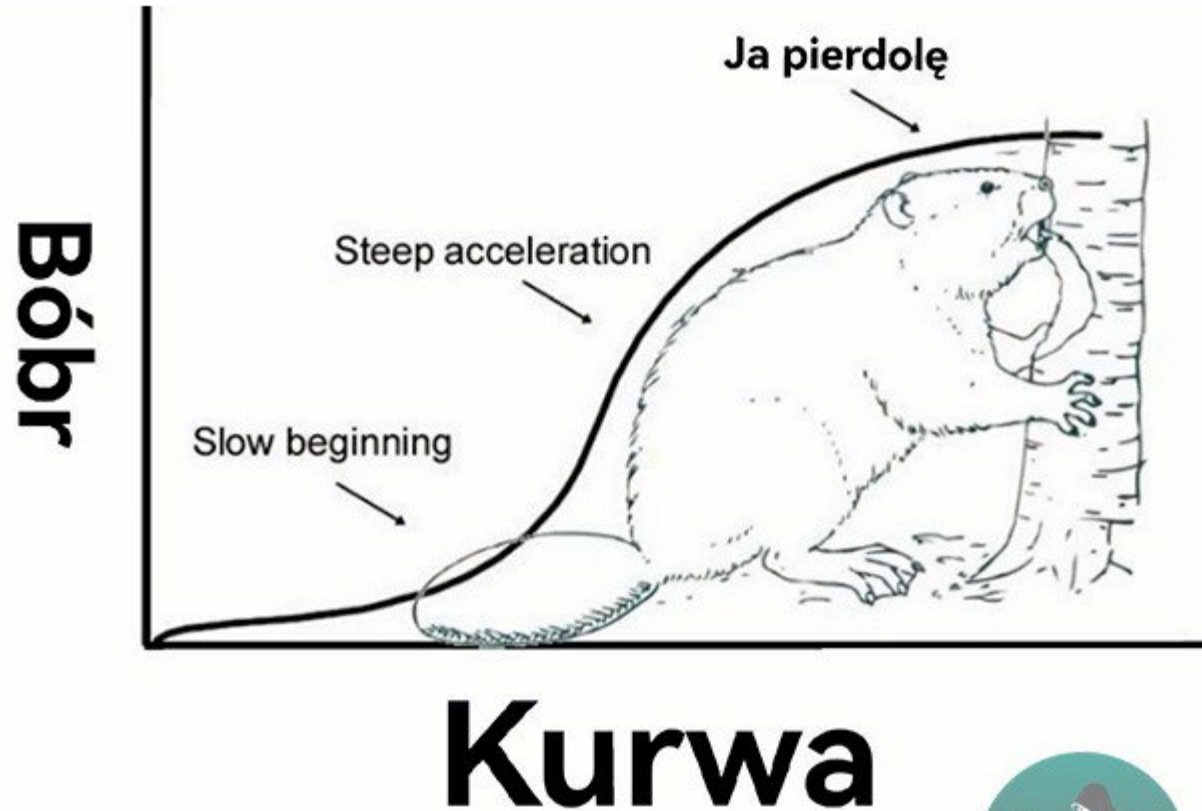
Сибирь I

Как начать свой путь в программировании

Мотивирующая прелюдия

А собственно зачем нам C++??

ТЫ НЕ ПОНИМАЕШЬ ЗАЧЕМ ОН ТЕБЕ НУЖЕН



**ЯЗЫК, КОТОРЫМ ТЫ
ПОЛЬЗОВАТЬСЯ НЕ БУДЕШЬ**

А если серьёзно

- C++ – лучший язык для начала программирования
- C++ активно используется в громаде современных проектов
- C++ можно запустить везде
- C++ может всё

Почему лучший?

В C++ представлены (почти) все используемые концепты программирования:

- Функциональное программирование
- Структурное программирование
- Объектно-ориентированное программирование
- Многопоточность
- Ручное управление памятью

Где используется?

- Android (NDK) и iOS
- Игрушки различные
- QT
- Python (внезапно)
- Операционные системы
- Прошивки и прочее низкоуровневое

Где можно запустить?

- *NIX и подобные
- Windows
- Прочие не *NIX-подобные
- Отсутствие операционной системы
- Распределённые ужасы

А что может?

- C++ - низкоуровневый ЯП (можно цепляться к железу)
- C++ имеет интерфейсы к операционной системе
- C++ имеет самый большой набор библиотек, в том числе C

**C++ - это круто, даже если вы
его не используете**

Наши инструменты

- GitHub
- VSCode
- Google

Приступим к установке

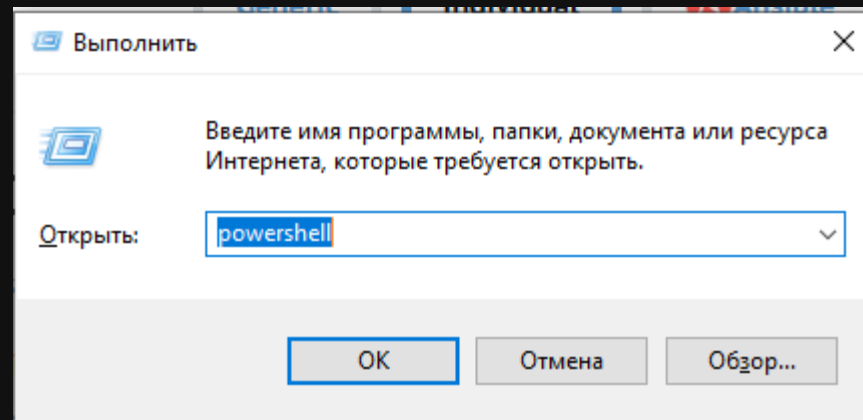
Если вы average windows slave fan

Проще будет поставить Linux (<3)

Превращаем винду в Linux

1. Устанавливаем пакетный менеджер chocolatey

Win + R – открывается окошко запуска программ



Запускаем программу от имени администратора
через Ctrl + Shift + Enter

Вводим следующие команды (можете загуглить установку chocolatey)

```
Get-ExecutionPolicy
```

```
# Если возвращается Restricted, то колдуем
```

```
Set-ExecutionPolicy Bypass -Scope Process
```

```
# Устанавливаем (это все одна строчка)
```

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
```

```
[System.Net.ServicePointManager]::SecurityProtocol =
```

```
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072;
```

```
ies ((New-Object System.Net.WebClient).DownloadString
```

```
('https://community.chocolatey.org/install.ps1'))
```


Как примерно выглядит

Try the new cross-platform PowerShell <https://aka.ms/powershell>

```
PS C:\Windows\system32> Get-ExecutionPolicy
```

Restricted

```
PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process
```

Execution Policy Change

The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the `about_Execution_Policies` help topic at <https://go.microsoft.com/fwlink/?LinkID=135170>. Do you want to change the execution policy?

[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y

```
PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).Download('https://community.chocolatey.org/install.ps1'))
```

Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)

Getting latest version of the Chocolatey package for download.

Not using proxy.

Getting Chocolatey from <https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2>.

Downloading <https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2> to C:\Users\ghost\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip

Not using proxy.

Extracting C:\Users\ghost\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\ghost\AppData\Local\chocolatey\chocoInstall

Installing Chocolatey on the local machine

Creating ChocolateyInstall as an environment variable (targeting 'Machine')

Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'

WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.

Restricting write permissions to Administrators

We are setting up the Chocolatey package repository.

The packages themselves go to 'C:\ProgramData\chocolatey\lib'

(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).

A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'

and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

chocolatey.nupkg file not installed in lib.

Attempting to locate it from bootstrapper.

PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...

WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\ghost\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.

Chocolatey (choco.exe) is now ready.

You can call choco from anywhere, command line or powershell by typing choco.

Run choco /? for a list of functions.

You may need to shut down and restart powershell and/or consoles

first prior to using choco.

Ensuring Chocolatey commands are on the path

Ensuring chocolatey.nupkg is in the lib folder

2. Через choco устанавливаем систему сборки meson (устанавливаем всегда из под администратора)

```
choco install meson
```

Как примерно выглядит

```
PS C:\Windows\system32> choco install meson
Chocolatey v2.2.2
Installing the following packages:
meson
By installing, you accept licenses for the packages.
Progress: Downloading meson 1.2.1... 100%

meson v1.2.1 [Approved]
meson package files install completed. Performing other installation steps.
The package meson wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

Downloading meson 64 bit
  from 'https://github.com/mesonbuild/meson/releases/download/1.2.1/meson-1.2.1-64.msi'
Progress: 100% - Completed download of C:\Users\ghost\AppData\Local\Temp\chocolatey\meson\1.2.1\meson-1.2.1-64.msi (
1 MB).
Download of meson-1.2.1-64.msi (13.91 MB) completed.
Hashes match.
Installing meson...
meson has been installed.
  meson may be able to be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of meson was successful.
Software installed as 'MSI', install location is likely default.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

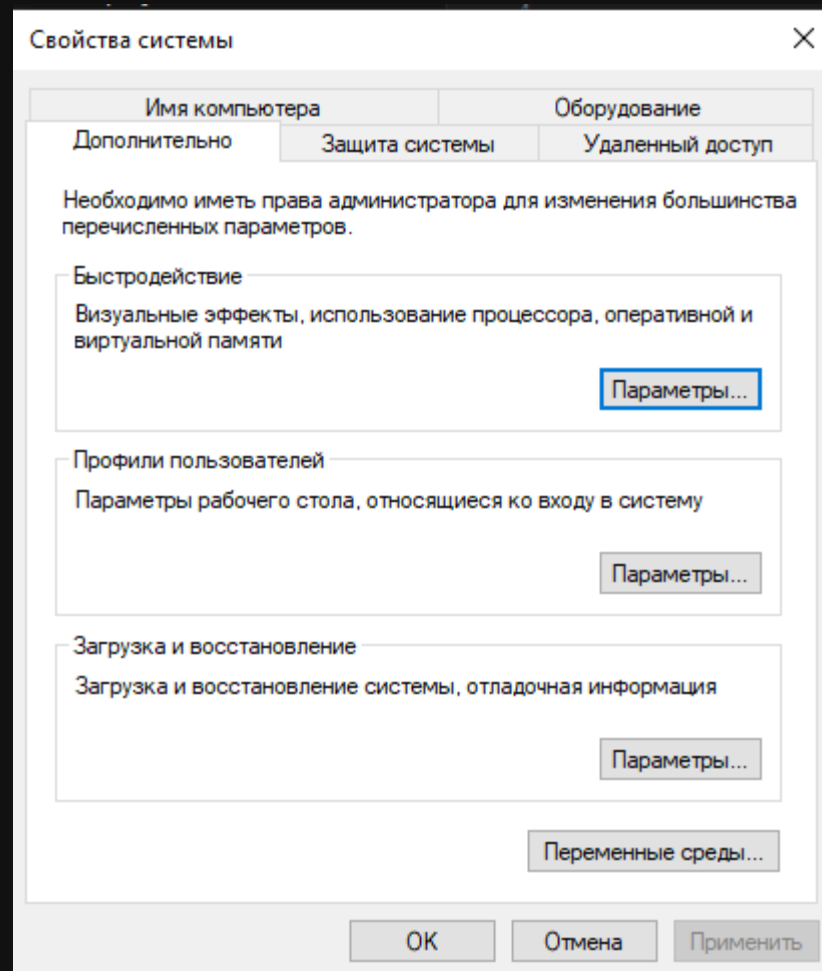
3. Через choco устанавливаем Git

```
choco install git
```

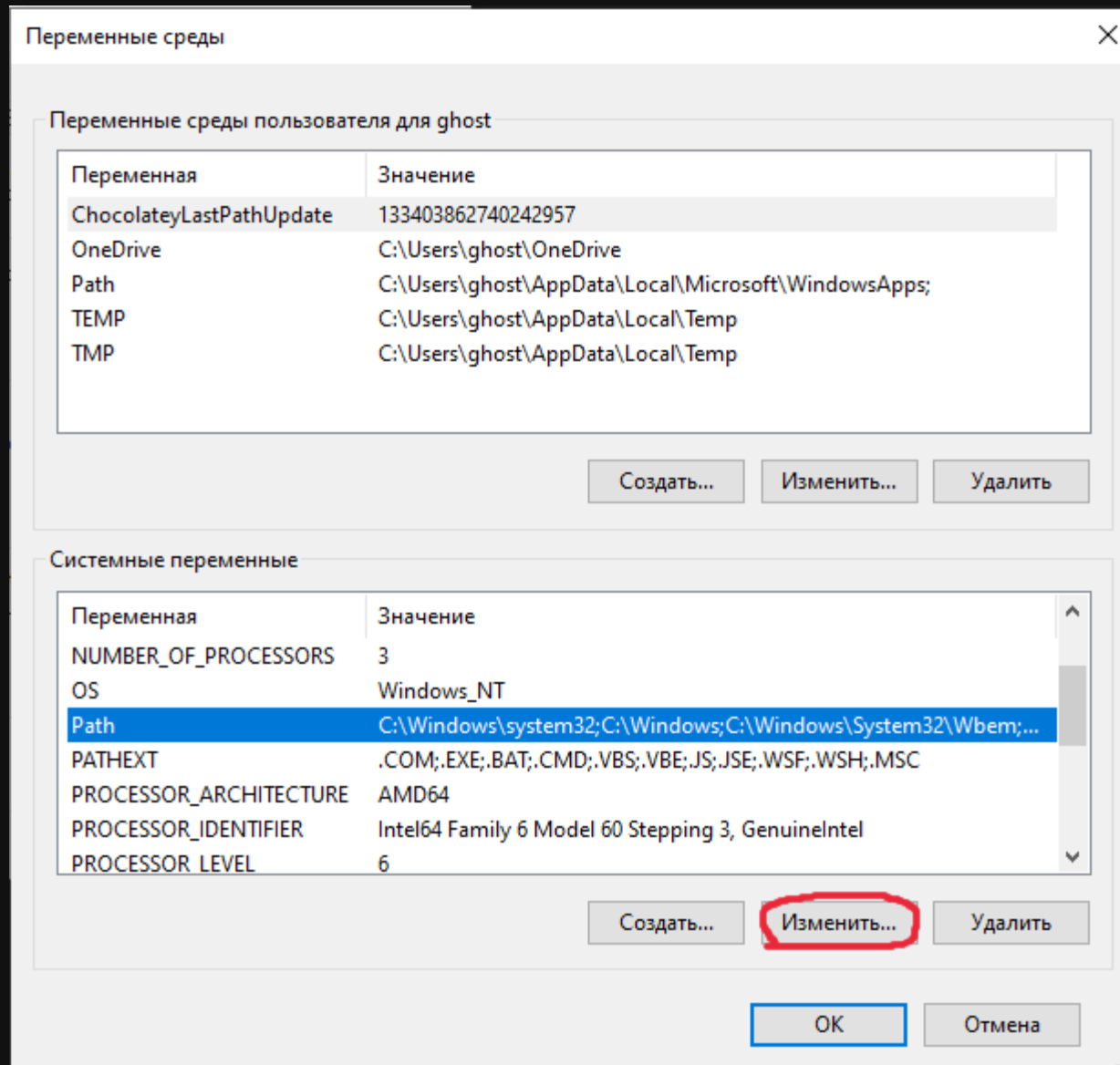
4. Устанавливаем MSYS2 (из интернета)

5. Настраиваем MSYS2 так, чтобы оно запустилось

1. Жмякаем Win + R
2. Вводим sysdm.cpl + enter
3. Открываем переменные среды



Залезаем в системные переменные PATH



**Создаем и вводим <адрес установки
msys2>\usr\bin**

у меня в примере Msys2 установлен в C:\soft
\msys64

Изменить переменную среды ✕

%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\soft\msys64\usr\bin
C:\ProgramData\chocolatey\bin
C:\Program Files\Meson\

Создать

Изменить

Обзор...

Удалить

Вверх

Вниз

Изменить текст...

ОК

Отмена

ОК

Отмена

6. Ставим VSCode (как угодно)

Если вы average Linux enjoyer

Снимаем шляпу

**Нужно поставить только meson и vscode
(обязательно сборку от microsoft)**

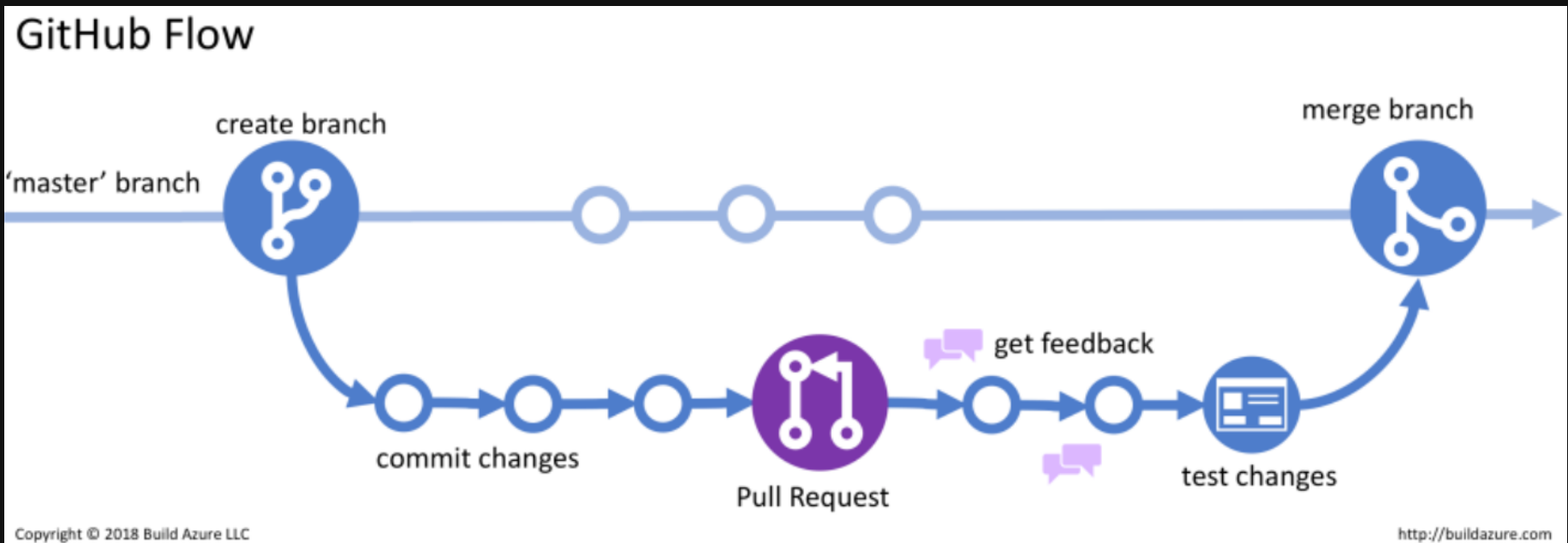
Делается это в одну строчку (пример для ubuntu)

```
sudo apt install meson
```

GitHub

И как с ним бороться

Как это концептуально работает



С Git мы работаем через терминал

Для Windows – cmd или powershell

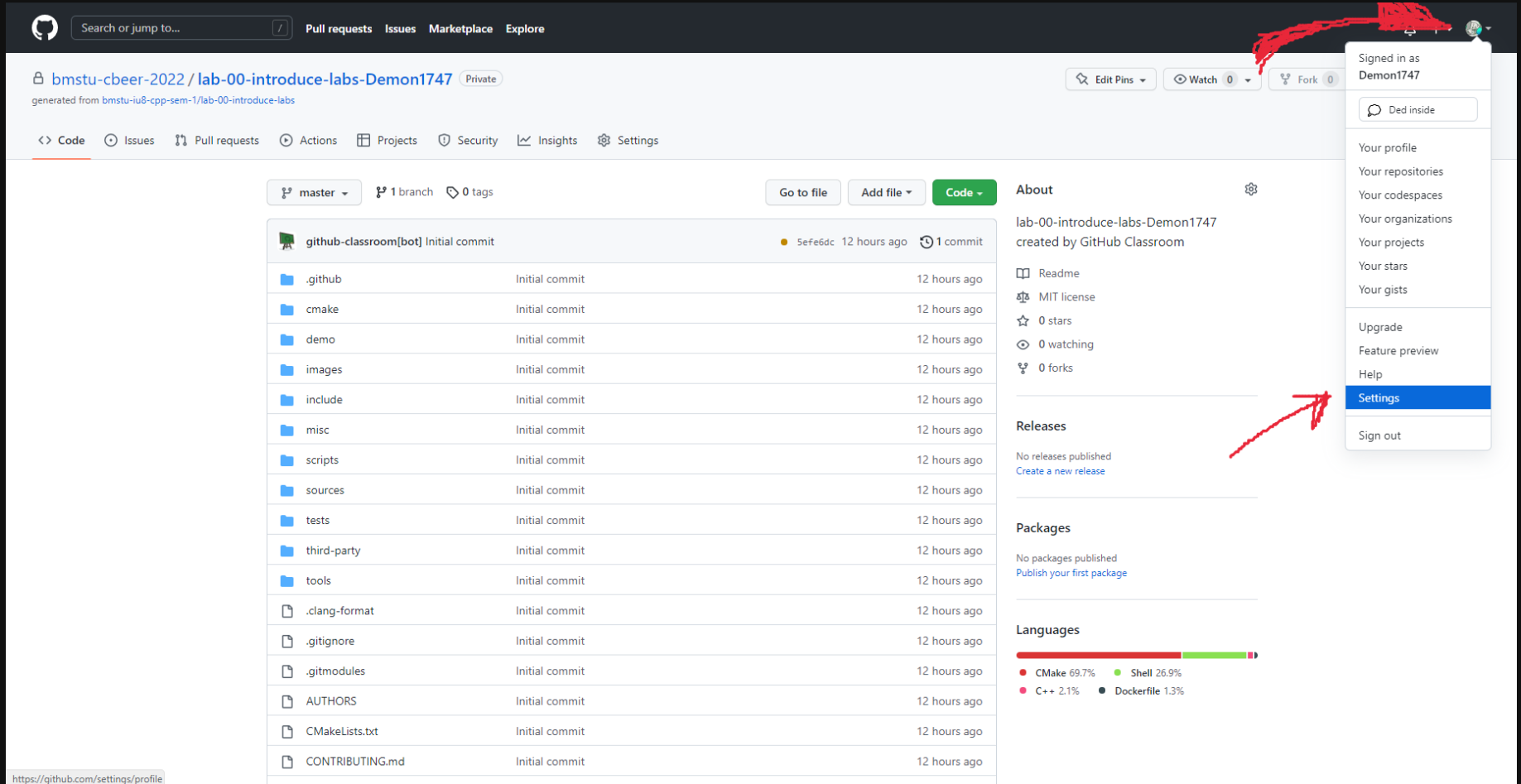
Генерируем ключ

```
ssh-keygen
```

И запоминаем содержание сгенерированного файла (в терминале можно с помощью команды `cat`):

```
cat ~/.ssh/id_rsa.pub
```

User settings



The screenshot shows the GitHub interface for a repository named `bmstu-cbeer-2022 / lab-00-introduce-labs-Demon1747`. The repository is private and was generated from `bmstu-iu8-cpp-sem-1/lab-00-introduce-labs`. The main content area displays a file tree with the following files and folders, all marked as 'Initial commit' 12 hours ago:

- `.github`
- `cmake`
- `demo`
- `images`
- `include`
- `misc`
- `scripts`
- `sources`
- `tests`
- `third-party`
- `tools`
- `.clang-format`
- `.gitignore`
- `.gitmodules`
- `AUTHORS`
- `CMakeLists.txt`
- `CONTRIBUTING.md`

On the right side, the 'About' section shows the repository was created by GitHub Classroom. Below it, the 'Releases' and 'Packages' sections indicate no releases or packages have been published. The 'Languages' section shows a bar chart with the following data:

Language	Percentage
CMake	69.7%
Shell	26.9%
C++	2.1%
Dockerfile	1.3%

A user menu is open in the top right corner, showing the user is signed in as `Demon1747`. The menu includes options like 'Ded inside', 'Your profile', 'Your repositories', 'Your codespaces', 'Your organizations', 'Your projects', 'Your stars', 'Your gists', 'Upgrade', 'Feature preview', 'Help', **Settings** (highlighted with a red arrow), and 'Sign out'.

<https://github.com/settings/profile>

Копаемся в настройках



Антон Климанов (zinstack625)

Your personal account [Switch to another account](#) ▼

Public profile

- Account
- Appearance
- Accessibility
- Notifications

Access

- Billing and plans ▼
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys**
- Organizations
- Enterprises
- Moderation ▼

Code, planning, and automation

- Repositories
- Codespaces
- Packages

Public profile

Name

Антон Климанов

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

koala.klimanov@gmail.com

✕ Remove

You can manage verified email addresses in your [email settings](#)

Bio

Student at BMSTU (information security)

You can @mention other users and organizations to link to them.

Pronouns

Don't specify

URL

https://t.me/zinstack625

Social accounts

[Link to social profile](#)

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys



SSH

Hermes

SHA256: Jb4Bo4P5UkMo9L3IS0wSN9fFlRAQzy6Qssa6utPhY

Added on Sep 26, 2022

Last used within the last 8 months — Read/write

Delete



SSH

Hades-android-dev

SHA256: YoeBE/vbkyu5ZdwU3wkfKuLDmgLvP1IB5N5fjZlSEJE

Added on Feb 11, 2023

Last used within the last 8 months — Read/write

Delete



SSH

anton@hades.zinstack.ru

SHA256: +WfwLDw6Z8sUX44k+UqIEW11gH2V1vSU5iu+Zcv7rIE

Added on Jul 22, 2023

Last used within the last 2 months — Read/write

Delete



SSH

Tartarus

SHA256: AdKeH9FXxkjqlK9lAwVedya6icolQMIxiudZy/g5W0

Added on Sep 11, 2023

Last used within the last week — Read/write

Delete

Вставляем свой ключик

Add new SSH Key

Title

Very key, much secure

Key type

Authentication Key

Key

ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBDG7eStQjOAZwH/tRBJtN108K93d9B0WaKNBj
ylZYQtONKvEOK6JLVZQIArz+DhOaqUBrKf9SFn+ZUtQLArnGLg=

Add SSH key

Находим свой репозиторий на Git

The screenshot shows the GitHub homepage interface. On the left sidebar, under the 'Recent Repositories' section, a list of repositories is displayed. The repository 'bmstu-cbeer-2022/lab-00-introduce-labs-Demon1747' is highlighted with a red rectangle. The main feed area shows a welcome message and a user profile for 'Stetics'.

Recent Repositories

- bmstu-iu8-g5-cpp-2020/01-lab-00-intro-Demon1747
- bmstu-cbeer-2022/lab-00-introduce-labs-Demon1747**

Recent activity

- bmstu-cbeer-2021-lab1/test_lab_01#9 roflapotsan
- bmstu-cbeer-2021-lab1/test_lab_01#3 mnogolapiy
- bmstu-cbeer-2021-lab1/test_lab_01#5 cr3at1v3annis
- bmstu-cbeer-2021-lab1/test_lab_01#4

Welcome to the new feed!

We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started.

[Send feedback](#)

Stetics
Who???

9 repositories 3 followers

[Follow](#)

zinstack625 published a release · 3d

Universe2022

Let's build from here

Join the global developer event for cloud, security, community, and AI.


Register today and get a 20% off early bird discount.


[Register now](#)

GitHub Copilot

Get suggestions for lines of code and entire functions in real-time

[Learn more about Copilot](#)





 **Demon1747** ▾

Recent Repositories

New

00

 bmstu-iu8-g5-cpp-2020/01-lab-00-intro-Demon1747

 **bmstu-cbeer-2022/lab-00-introduce-labs-Demon1747**

Recent activity

Копируем ссылку

00-introduce-labs-zinstack625

Private

Edit Pins

Watch 0

Fork 0

bmstu-iu8-cpp-sem-1/lab-00-introduce-labs

1 branch 0 tags

Go to file

Add file

<> Code

About

01-lab-00-intro
created by GitHub

Readme

MIT license

Activity

0 stars

0 watching

0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

-classroom[bot] Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

Initial commit

last year

last year

last year

last year

last year

Local

Codespaces

Clone

HTTPS

SSH

GitHub CLI

git@github.com:bmstu-cbeer-2022/01-lab-00

Copied!



Use a password-protected SSH key.

Download ZIP

Code 55% faster with AI pair programming.

Buy Copilot for Business

[Don't show again](#)

Открываем терминал и переходим в папку, в которую хотим клонировать репозиторий

```
# Пример команды перехода в папку в терминале  
cd /home/projects
```

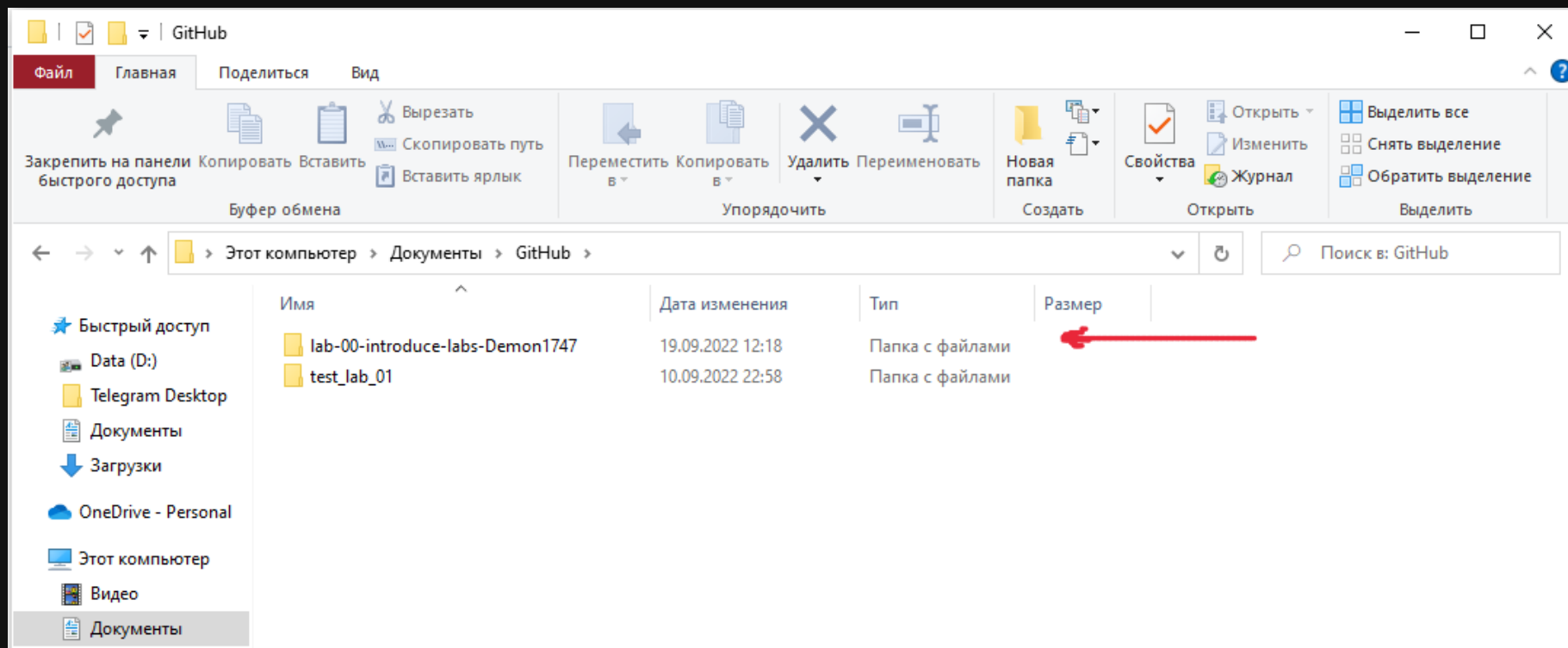
Клонируем репозиторий

```
# Данной командой  
git clone <ссылка>
```

```
# Пример
```

```
anton@tartarus:~/Code/ISCRA/CBeer> git clone git@github.com:  
bmstu-cbeer-2023/01-lab-00-introduction-CoolUser.git  
Cloning into '01-lab-00-introduction-CoolUser'...  
remote: Enumerating objects: 181, done.  
remote: Counting objects: 100% (181/181), done.  
remote: Compressing objects: 100% (155/155), done.  
remote: Total 181 (delta 30),  
reused 166 (delta 21), pack-reused 0  
Receiving objects: 100% (181/181), 33.53 MiB |  
784.00 KiB/s, done.  
Resolving deltas: 100% (30/30), done.
```


Магия



**Теперь можно писать
КОД**

VSCode

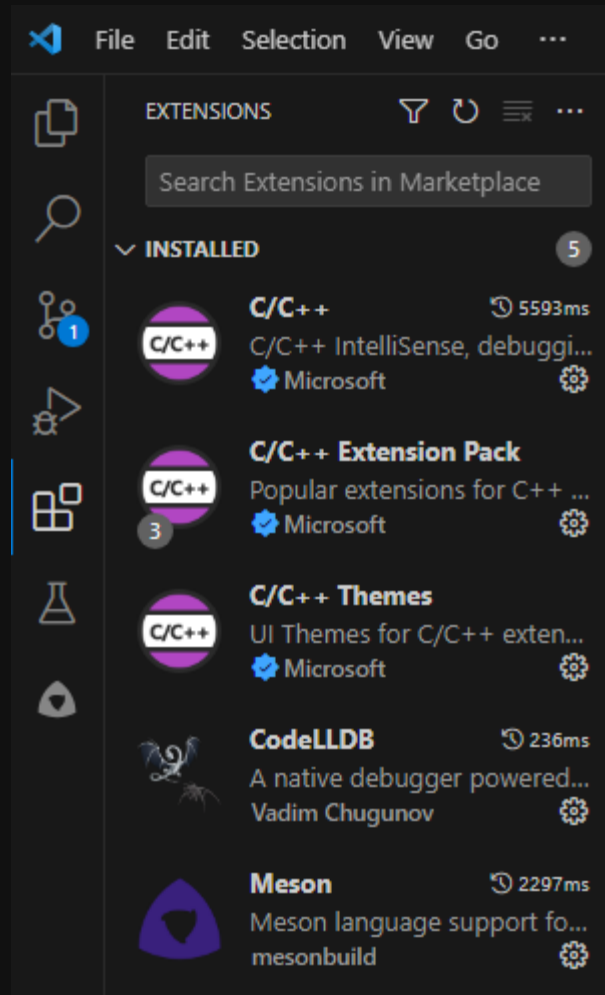
(или другой IDE/текстовый редактор)

Орудие, несущее свет

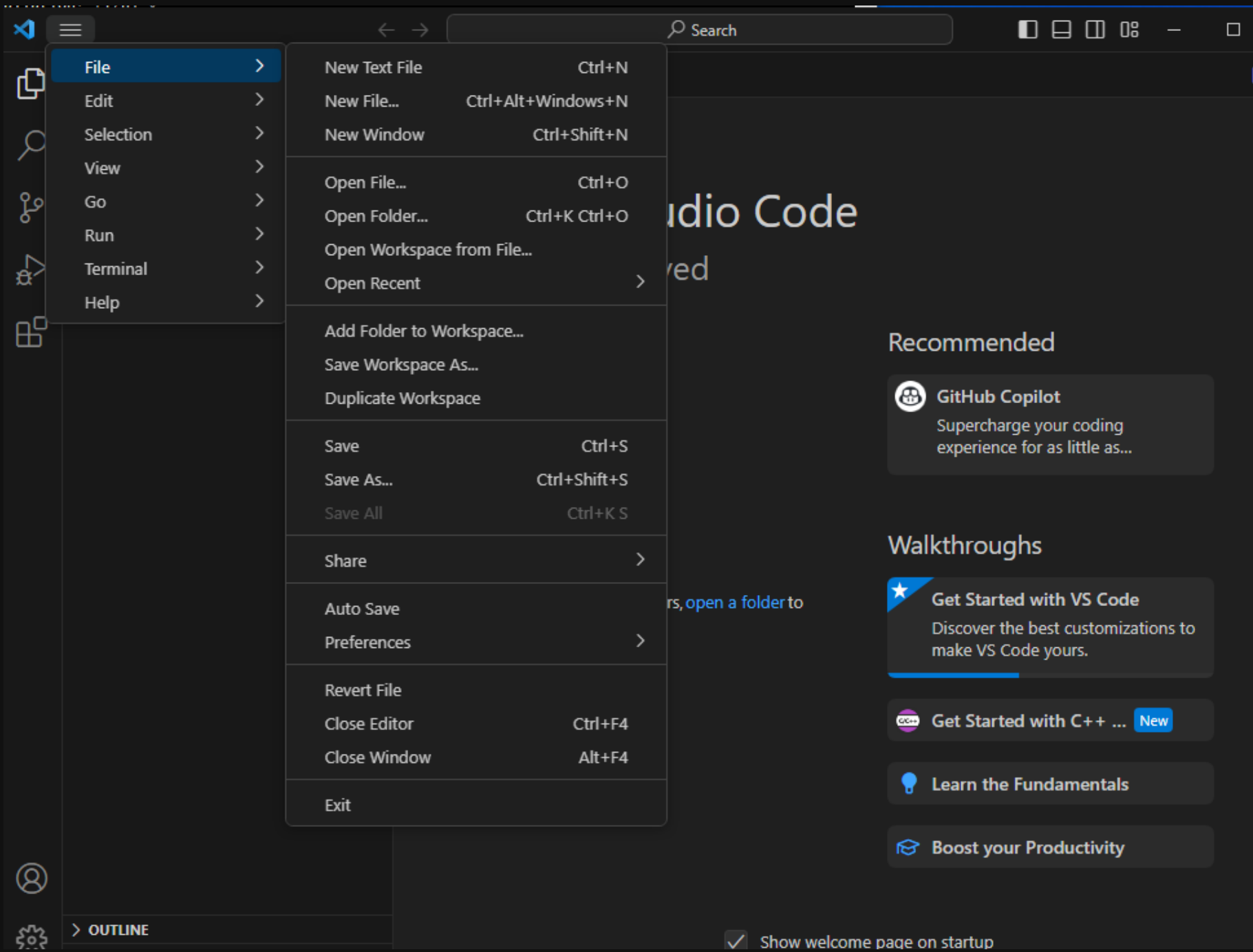
Интересные плагины, с которыми хочется жить

- Meson (от mesonbuild)
- CodeLLDB (от vadimcn)
- WSL (для Windows)

Ставим необходимые плагины



Открываем папку с репозиторием



VS Code

☰

← →

Search

🔍

📄

🔍

🔗

🔧

📦

File

Edit

Selection

View

Go

Run

Terminal

Help

New Text FileCtrl+N

New File...Ctrl+Alt+Windows+N

New WindowCtrl+Shift+N

Open File...Ctrl+O

Open Folder...Ctrl+K Ctrl+O

Open Workspace from File...

Open Recent>

Add Folder to Workspace...

Save Workspace As...

Duplicate Workspace

SaveCtrl+S

Save As...Ctrl+Shift+S

Save AllCtrl+K S

Share>

Auto Save

Preferences>

Revert File

Close EditorCtrl+F4

Close WindowAlt+F4

Exit

idio Code

ved

Recommended

👤

GitHub Copilot

Supercharge your coding experience for as little as...

Walkthroughs

★

Get Started with VS Code

Discover the best customizations to make VS Code yours.

🔗

Get Started with C++ ...

New

💡

Learn the Fundamentals

🎓

Boost your Productivity

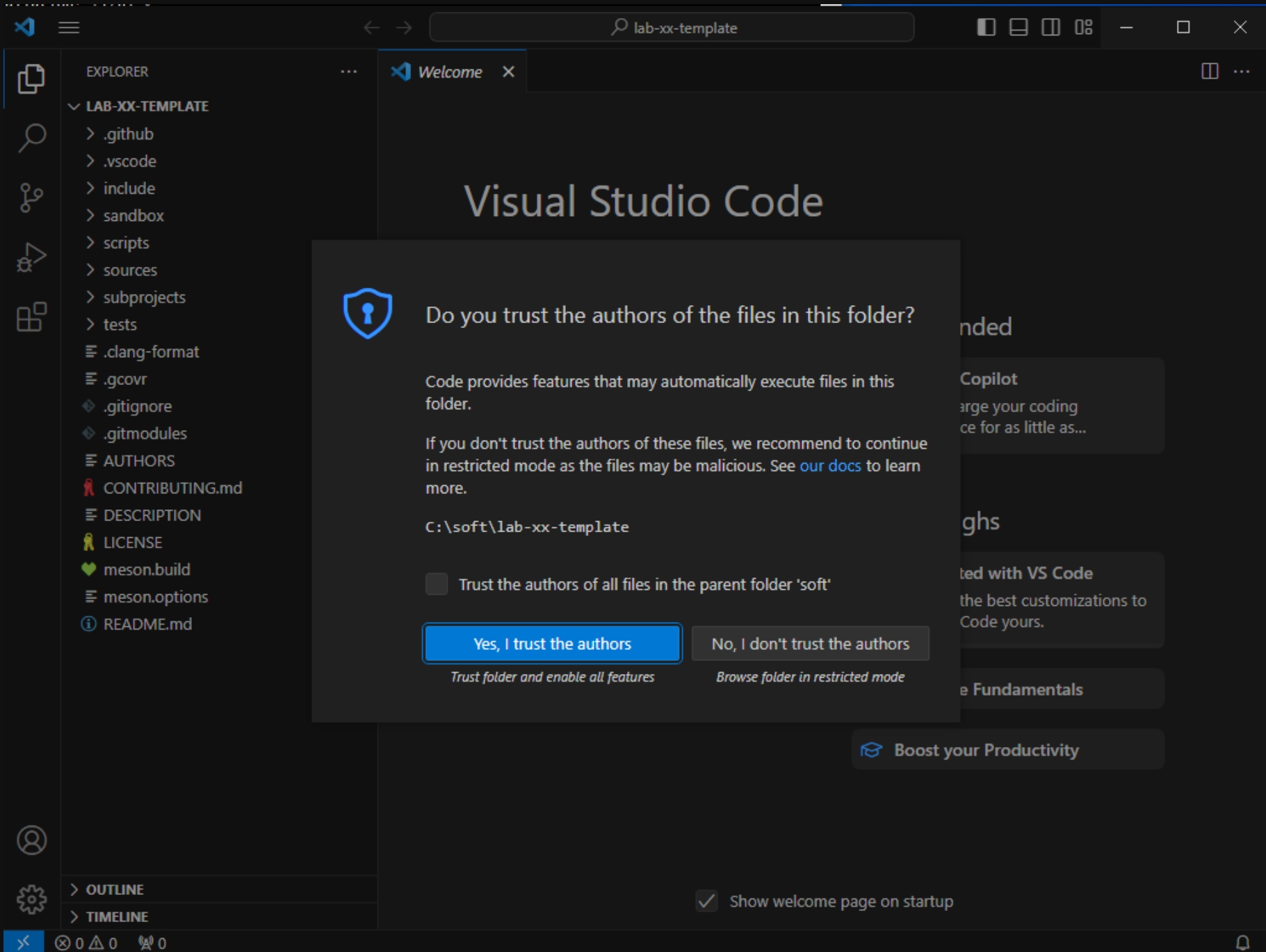
👤

OUTLINE

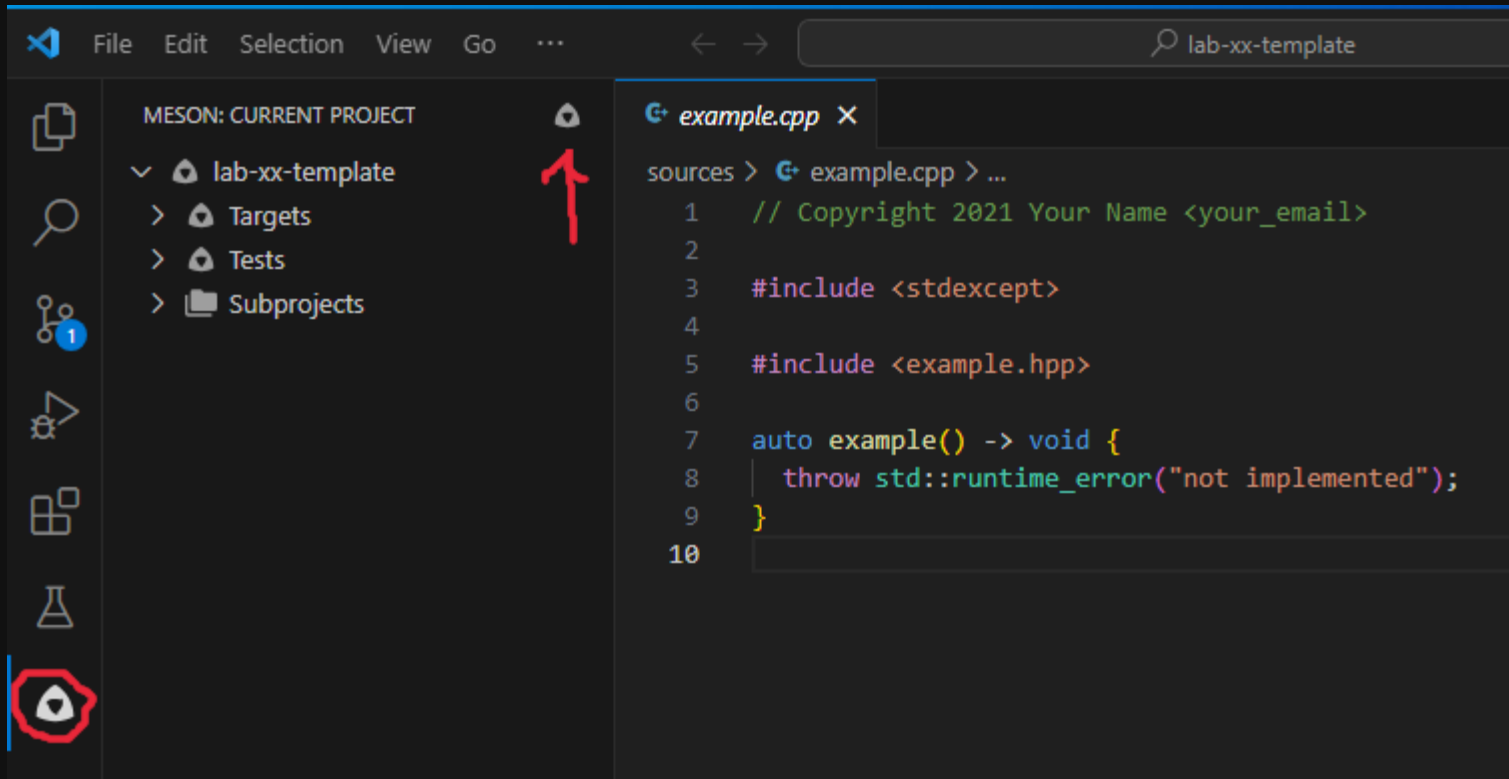
🔧

☑ Show welcome page on startup

Доверяем



Запускаем конфигурацию проекта, если само не сработало




Устройство лабораторной

Name ▲

 include

Заголовочные файлы

 sandbox

Тут поиграться (не проверяем)

 scripts


Тут наша автоматика

 sources

Главное поле боя (проверяемый код)

 subprojects

Пока не трогаем

 tests


Как проверяем

 AUTHORS

 CONTRIBUTING.md

 DESCRIPTION

 LICENSE

 meson.build

 meson.options

 README.md

Написали код в новом файле?

- `sources/` - внесите его в `sources/meson.build`
- `headers/` - ничего делать не надо

Как устроена типичная программа

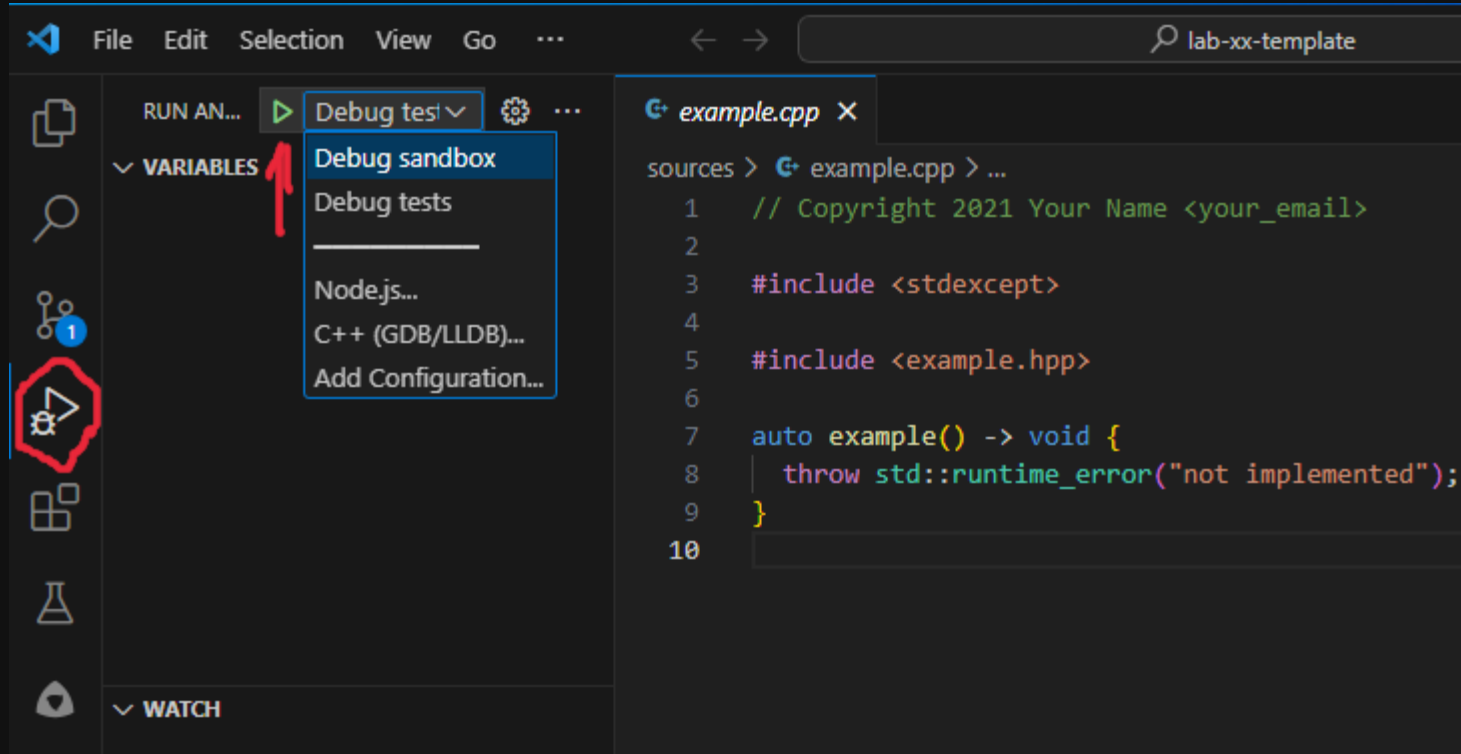
```
#include <библиотека>

int main() {
    // здесь пишем код
    return 0;          // Возвращаемое значение
}
```

Первые команды

```
std::cout << "Текст";           //вывод текста на экран
std::cout << "Welcome to the club, partner";
std::cout << std::endl;         // переход на следующую строку
std::cout << "Some text" << std::endl;
```

Запускаем код и смотрим, что получилось



**Самое время показать свое
детище миру**

Загрузка на GitHub

или как не испортить ветку master

Предварительная настройка

```
git config --global user.name "Your Name"  
git config --global user.email "your@email.tld"
```

Иначе git будет ругаться на git commit

Вот они слева направо

```
# переход в директорию
cd ./dir_name
# создать новую ветку (предпочтительно wr/lab)
git branch <название ветки>
# переход на другую ветку (только что созданную)
git switch <название ветки>
# выводит изменения в репозитории
git status
# добавление файла в коммит
git add <имя файла>
# подготовка коммита
git commit -m "<Комментарий>"
# отправляем на git
git push origin <название ветки>
```

Это не взлом Пентагона

```
anton@tartarus:~/CBeer/examble-repo> git branch wp/lab
anton@tartarus:~/CBeer/examble-repo> git switch wp/lab
Switched to branch 'wp/lab'
anton@tartarus:~/CBeer/examble-repo> git add new-file
anton@tartarus:~/CBeer/examble-repo> git commit -m 'lab done'
[wp/lab 6c479e6] lab done
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 new-file
anton@tartarus:~/CBeer/examble-repo> git push origin wp/lab
```

Еще немного...

wp/lab had recent pushes 2 minutes ago

Compare & pull request

github-classroom[bot] Initial commit 14 hours ago 1 commit

.github	Initial commit	14 hours ago
cmake	Initial commit	14 hours ago
demo	Initial commit	14 hours ago
images	Initial commit	14 hours ago
include	Initial commit	14 hours ago
misc	Initial commit	14 hours ago
scripts	Initial commit	14 hours ago
sources	Initial commit	14 hours ago
tests	Initial commit	14 hours ago
third-party	Initial commit	14 hours ago
tools	Initial commit	14 hours ago
.clang-format	Initial commit	14 hours ago
.gitignore	Initial commit	14 hours ago
.gitmodules	Initial commit	14 hours ago
AUTHORS	Initial commit	14 hours ago

About

lab-00-introduce-labs-Demon1747
created by GitHub Classroom

Readme
MIT license
0 stars
0 watching
0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

CMake	69.7%	Shell	26.9%
C++	2.1%	Dockerfile	1.3%

Создаем Pull request

bmstu-cbeer-2022 / lab-00-introduce-labs-Demon1747 (Private)

generated from bmstu-iu8-cpp-sem-1/lab-00-introduce-labs

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ← compare: wp/lab ✓ Able to merge. These branches can be automatically merged.

Название pull-request

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow its [contributing guidelines](#).

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use [Closing keywords](#) in the description to automatically close issues

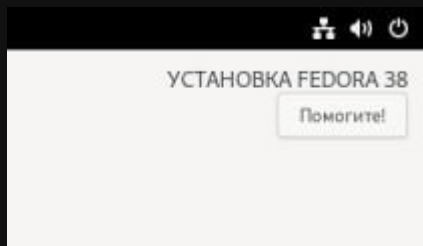
Helpful resources
[Contributing](#)
[GitHub Community Guidelines](#)

**Отправляем на проверку боту в любой точке
mattermost**

`/check_me https://github.com/bmstu-cbeer-
2023/01-lab-00-introduction-CoolUser/pulls/1`

“у меня что-то не работает, что делать?”

Спросить. Мы отвечаем на вопросы. Честно!



Затруднения на данном этапе – совершенно нормально, так что лучше не биться головой об стенку слишком долго, а разбиться, но чуть-чуть, и посоветоваться с обществом на курсе, наверняка подобные проблемы уже встречались :)

*У программиста нет push в master,
только pull request*

© Dema