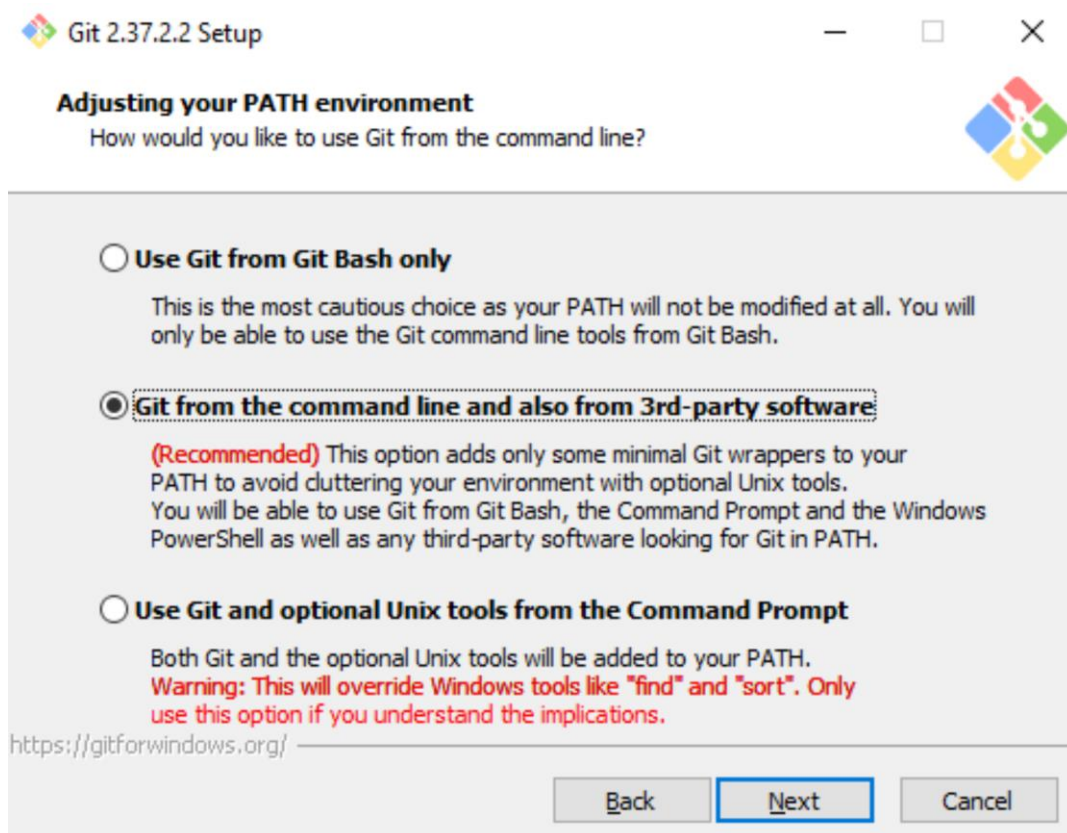


# Инструкция по работе с Git и GitHub

## Подготовка к работе

1. Для начала нужно установить Git на ваш компьютер.

Во время установки следует обратить на некоторые этапы и выбрать настройки как на картинках.



**Choosing the SSH executable**

Which Secure Shell client program would you like Git to use?

☒ **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

☐ **Use external OpenSSH****NEW!** This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.<https://gitforwindows.org/>

Back

Next

Cancel

**Configuring the line ending conversions**

How should Git treat line endings in text files?

☒ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

☐ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

☐ **Checkout as-is, commit as-is**

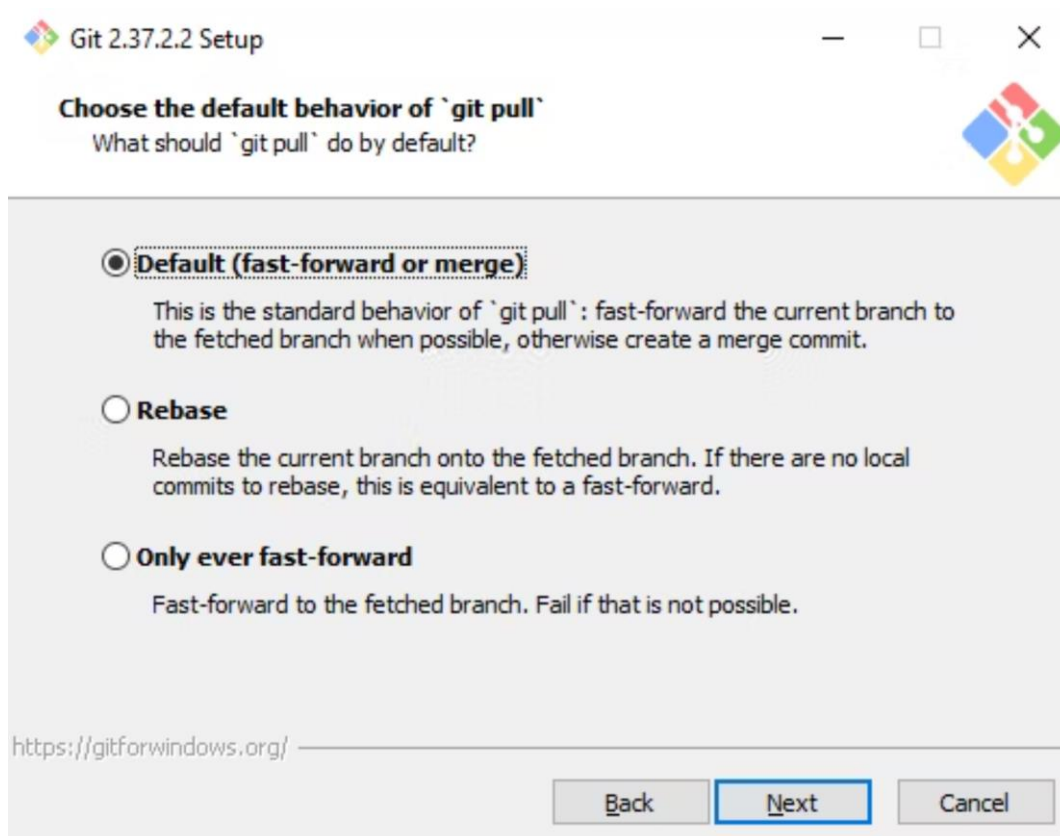
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

Back

Next

Cancel



2. Нужно настроить Git, для этого откройте терминал и введите, заменяя данные на свои:  

```
git config --global user.name "Ivan Ivanov"
```

```
git config --global user.email ivan.ivanov@gmail.com
```
3. Теперь зарегистрируйтесь на [GitHub](https://github.com): задайте логин, почту и придумайте пароль. После создания аккаунта не забудьте проверить почту и подтвердить её.
4. GitHub поддерживает безопасность за счет сетевых протоколов HTTPS и SSH. Работать будем, используя SSH. Нужно создать пару ключей (публичный и приватный), используя терминал. Для этого нужно:
  - 4.1. Открыть терминал.
  - 4.2. Ввести: 

```
ssh-keygen -t ed25519 -C your_email@example.com
```

  
(your\_email@example.com нужно заменить на вашу почту)

4.3. После этого следует нажать Enter, чтобы файл сохранился в папку по умолчанию.

4.4. Далее вам будет предложено установить пароль для использования ключа. Можно пропустить и этот пункт, нажав Enter.

```
PS C:\WINDOWS\system32> ssh-keygen -t ed25519 -C "rjtsn@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\rjtsn/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\rjtsn/.ssh/id_ed25519
Your public key has been saved in C:\Users\rjtsn/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:.....
The key's randomart image is:
+--[ED25519 256]--+
|      oo. .++|
|     .E . .+|
|    . =+o . +|
|   . =+.o+o|
|  . S = .o.=o|
|   = + . X|
|      X*|
|     .+X|
|      oX|
+-----[SHA256]-----+
PS C:\WINDOWS\system32>
```

5. Теперь нужно добавить приватный ключ в ssh agent, введя в терминал следующие команды:

5.1. *Start-Service ssh-agent*

5.2. *ssh-add ~/.ssh/id\_ed25519*

```
+-----[SHA256]-----+
PS C:\WINDOWS\system32> Start-Service ssh-agent
PS C:\WINDOWS\system32> ssh-add c:/Users/rjtsn/.ssh/id_ed25519
Identity added: c:/Users/rjtsn/.ssh/id_ed25519 (rjtsn@gmail.com)
```

6. Далее нужно добавить публичный ssh ключ в аккаунт GitHub:

6.1. Введите в консоли, чтобы скопировать публичный ключ:

*cat ~/.ssh/id\_ed25519.pub | clip*

6.2. Зайдите на GitHub.

6.3. В верхнем правом углу нажмите на фото профиля, затем нажмите «Settings» (Настройки).

6.4. В боковом меню в секции «Access» выберите «SSH and GPG keys».

6.5.Нажмите «New SSH key» или «Add SSH key».

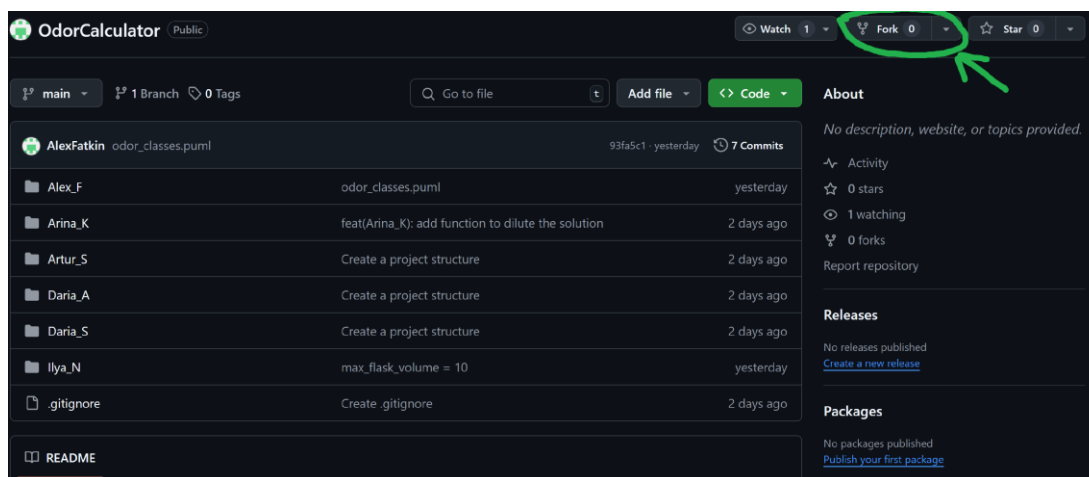
6.6.В поле «Title» дайте название для нового ключа, например «Personal laptop».

6.7.В поле «Key» вставьте ваш скопированный публичный ключ.

6.8.Нажмите «Add SSH key».

6.9.Если требуется, подтвердите доступ к аккаунту GitHub.

7. Теперь нужно зайти в наш проект на GitHub и нажать кнопку «Fork».



8. Далее нужно зайти в свои репозитории и выбрать там форк OdorCalculator.

9. Нажимаем на зеленую кнопку «Code», выбираем «SSH» и копируем строку, начинающуюся с `git@github.com`

10. На компьютере выбираем/создаем папку, в которой будут храниться проекты. Нажимаем на нее правой кнопкой мыши и выбираем «Git Bash Here» или «Открыть в терминале».

11. Клонировем репозиторий к себе на компьютер введя команду, заменив `git@github.com:project` на то, что мы скопировали ранее:

*`git clone git@github.com:project`*

12. Нужно настроить синхронизацию форка и главного репозитория, введя команду:

```
git remote add upstream git@github.com:AlexFatkin/OdorCalculator.git
```

## Работа с проектом

13. Нам нужно выбрать на компьютере папку с проектом OdorCalculator, нажать правой кнопкой мыши по ней и выбрать «Git Bash Here» или «Открыть в терминале».

14. Теперь нам нужно перейти в ветку main, вводим команду:

```
git checkout main
```

15. Каждую новую разработку, функцию нужно делать в отдельной ветке. Перед тем, как создавать ветку, нужно синхронизировать нашу ветку main. Для этого вводим в терминал: *git pull --rebase upstream main*

16. Теперь нужно создать ответвление от нашей главной ветки main, введя в терминале команду, где после -b вводим название новой ветки, которое будет отражать то, что мы там сделаем:

```
git checkout -b название_ветки
```

17. Теперь в папке с вашим именем можно создавать файлы, исправлять их, писать код и делать прочую работу. Чтобы сделать коммит, нужно для начала добавить файлы, которые мы хотим включить в коммит. Для этого вводим:

```
git add ./Название_вашей_папки
```

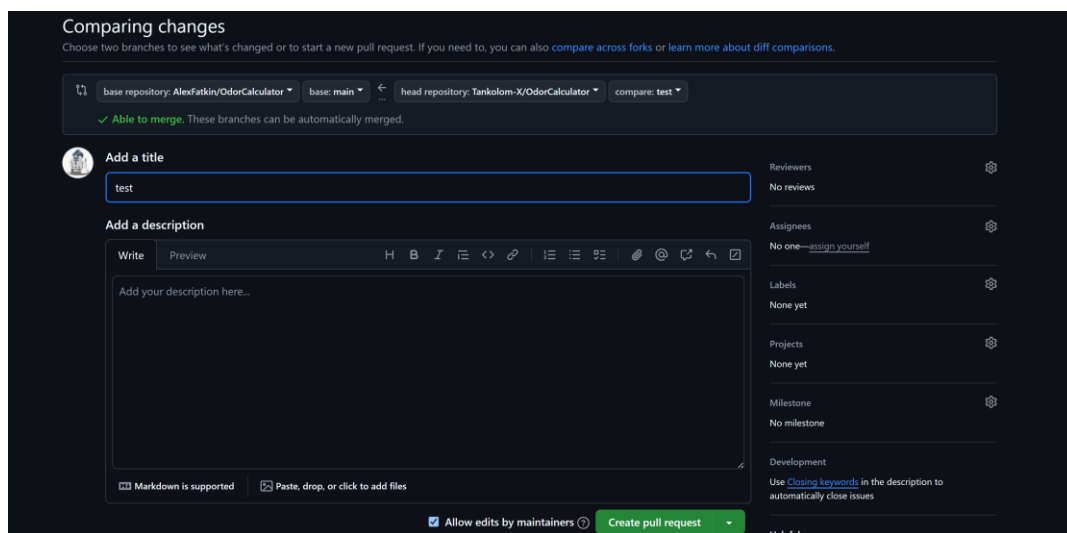
18. Теперь можно создать коммит, введя команду:

```
git commit -m "Название коммита"
```

19. Теперь нужно отправить коммит на свой репозиторий командой:

```
git push origin название_ветки
```

20. Теперь заходим в GitHub в свой форк и, чтобы создать pull request, нажимаем кнопку «Compare & pull request». Далее появится следующее окно.



Вводим название нашего pull request, выбираем ветку откуда и куда мы хотим внести наши изменения (из нашей созданной в форке ветки в ветку main проекта) и нажимаем «Create pull request».

21. Чтобы начать делать что-либо новое, возвращаемся к пункту 13 и повторяем действия приведенные в разделе «Работа с проектом».