

# Удаленная сдача задач через Github

Илья Булгаков, Валерий Сенотов, Виталий Ерошин

Московский физико-технический институт

Москва, 2023

# Содержание

- 1 Настройка удаленного репозитория на Github
- 2 Как работать локально?
- 3 Как сделать pull request?
- 4 Распространенные ошибки

# Настройка удаленного репозитория на Github

- 1 Зайдите на Github.com
- 2 Зарегистрируйтесь, выберите логин (будет удобно, если он будет совпадать с логином в констесте).
- 3 Создайте репозиторий с именем algorithms-mipt, выберите опцию Приватный.
- 4 При создании не забудьте добавить файл .gitignore для C++!
- 5 Добавьте семинариста (Вкладка Settings -> Collaborators) с админскими правами. Укажите логин вашего семинариста.
- 6 Семинарист примет приглашение по почте

Важно:

- 1 Вся работа в семестре должна вестись в рамках этого репозитория. Новые создавать не нужно.
- 2 Без пройденного ревью на Github задачи с кодревью не засчитываются

# Как привязать ключ к Github?

- 1 Для начала нужно создать пару ключей для ssh - публичный и приватный. Это делается в терминале через `ssh-keygen`. Вместо `your_email@example.com` следует указать почту, привязанную к вашему Github аккаунту.

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

- 2 По умолчанию ключи сохраняются в папке `/.ssh`. Посмотрим на публичный ключ, который мы сгенерировали.

```
cat ~/.ssh/id_ed25519.pub
```

- 3 Теперь нужно привязать ключ к нашему аккаунту на Github. Это делается тут: <https://github.com/settings/ssh/new>.

# Как работать в репозитории?

Нужно установить Git на ваш компьютер. После этого вы сможете работать локально и синхронизировать ваши правки с удаленным репозиторием. Работа с git идет через Terminal (Linux, MacOS) или PowerShell (Windows).

- ❶ Сделайте `git clone <ссылка из Github>` (кнопка Code -> SSH)
- ❷ `master` - ветка, куда вливается подтверждённый семинаристом код.
- ❸ Для каждой задачи отдельная ветка.  
Имя ветки `moduleX-taskY`.
- ❹ Рекомендуется делать для каждой новой задачи новую папочку (например, `1_1`, `1_2` и т.д.), тогда не будет коллизий при вливании в `master`
- ❺ Стандартный сценарий при настроенном репозитории (команды запускаются из папки с репозиторием):
  - ❶ `git checkout master` \$ синхронизовать репозиторий
  - ❷ `git checkout -b <название или номер задачи>` \$ создать ветку
  - ❸ делаете изменения, сдаете в контексте
  - ❹ `git add <файл>` \$ внести файлы в коммит
  - ❺ `git commit -m <сообщение>` \$ добавить коммит в ветку
  - ❻ `git push <имя репозитория> <имя ветки>` \$ отправить коммит на github

# Как сделать pull request?

- ❶ Когда всё готово - идете во вкладку Pull requests - New pull request.
- ❷ Указываете вашу ветку в качестве источника (compare). master в качестве назначения (base)
- ❸ Нажмите Create pull request
- ❹ Справа выберите в reviewers профили семинаристов
- ❺ Названия pull request'ов должны быть по шаблону:  
[Имя Фамилия] <номер модуля>-<номер задачи>-<название задачи>.  
Например,  
[Алексей Петров] Модуль1-Задача1-Бинарный поиск
- ❻ В описании (Description) pull request'а должно быть:
  - 1) Ссылка на первую посылку (уложились в дедлайн)
  - 2) Ссылка на последнюю посылки (те правки, которые вы сдаете)
- ❼ Нажмите Create pull request

# Как сделать pull request?

- 1 Семинаристы делают ревью, пишут замечания
- 2 Студенты исправляют замечания, тестируют, коммитают исправленный код в ту же ветку, делают push. Обязательно обновить посылку. Pull request обновляется автоматически после сделанного push.
- 3 Семинарист ставит коду "approved"
- 4 Студент нажимает кнопку "merge чтобы правки были влиты

# Распространенные ошибки

- ❶ После того, как вы закончили работу с одной задачей и хотите начать работу над новой, - не забудьте переключиться на master (`git checkout master`)! Сделайте `git pull` в мастер.



# Полезные ссылки I



Pro Git

<https://git-scm.com/book/ru/v1>



Установка Git

<https://git-scm.com/downloads>