

# Обзор системы контроля версий Git

## Практикум на ЭВМ 2017/2018

Солоткий Михаил

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

14 ноября 2017 года

# Что такое система контроля версий и зачем она нужна.

- Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.
- Вообще, если, пользуясь СКВ, вы всё испортите или потеряете файлы, всё можно будет легко восстановить.

# Преимущества Git.

- Быстрое переключение между версиями (в отличие от CVS, Subversion, Perforce, Bazaar);
- Git - распределённая система (в отличие от CVS, Subversion и Perforce);
- Git следит за безопасностью данных.

# Установка git - утилиты для системы Git.

- Windows

Скачиваем .exe инсталлятор с <http://msysgit.github.com/>

- Mac

Можно скачать графический инсталлятор.

<http://sourceforge.net/projects/git-osx-installer/>

- Linux

```
$ yum install git-core # Fedora
```

```
$ apt-get install git # Debian or Ubuntu
```

# Как начать контроль версий с git.

- Создать пустой репозиторий с 0 зафиксированных изменений.

```
$ git init
```

- Скопировать существующий репозиторий со всей историей.

```
$ git clone https://github.com/esokolov/ml-course-msu.git
```

# Коммиты.

- Добавляем отслеживание файлов.

```
$ git add . # Для всех файлов
```

```
$ git add README # Для файла README
```

- Фиксируем изменение.

```
$ git commit -m 'first commit'
```

```
[master (root-commit) 11730f0] first commit
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 README
```

# Ветвление.

- Ветка - просто указатель на последний коммит.
- Создание ветки branch и работа в ней.

```
$ git checkout -b branch
```

```
$ vim index.html
```

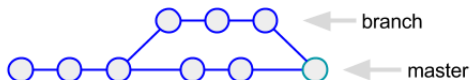
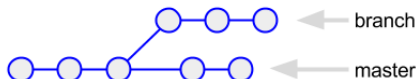
```
$ git commit -a -m 'added a new footer [issue 53]'
```

# Ветвление.

- Слияние ветки branch с веткой master.

```
$ git checkout master
```

```
$ git merge branch
```



- Удаление ветки branch.

```
$ git branch -d branch
```



## Откат до какого-то из прошлых коммитов.

- Небезопасный способ:

```
$ git checkout -- README
```

- Безопасный способ:

```
$ git log # Чтобы узнать hash коммита
```

```
$ git checkout <commit hash>
```

```
$ git checkout -b new_branch
```

- Больше самых разных способов:

<https://ru.stackoverflow.com/questions/431520/Как-вернуться-к-более-раннему-коммиту>

# Самое важное, что нужно знать о git:

- Как инициировать контроль версий;
- Как делать коммиты;
- Как переключиться на другую версию;
- Как создавать, переключаться, удалять и сливать ветки;
- Как работать с удалённым сервером;
- Как разрешать конфликты.

Книга по пользованию git:

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