

# Система контроля версий

Михаил Шавкунов

Руководитель: Антон Афанасьев  
СПб АУ РАН

31 мая, 2016

# Проблема организации файлов

- Представим ситуацию: над большим проектом трудится много людей. Каждый вносит изменения в файл. Как полезные так и бесполезные. Хотелось бы знать, кто и когда их внес.
- Иногда в проект вносятся некорректные или попросту ненужные изменения. В этом случае возникает необходимость "откатиться" на предыдущую версию проекта.
- На помощь в решении этих проблем и приходит система контроля версий.

# Цель работы

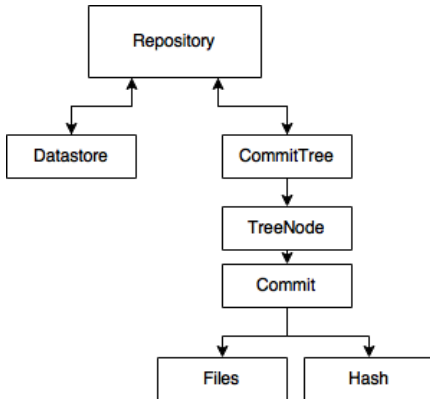
- Моя задача была написать свою систему контроля версий.
- Нужно было, чтобы работали хотя бы базовые команды: добавить файл(ы) в репозиторий, возможность их откатывать.

# Подзадачи

- 1 Разобраться с уже имеющимся кодом и взять его за основу.
- 2 Разобраться с применением таких библиотек в проекте, как `boost::filesystem`, `boost::serialization`, `CMake`, `OpenSSL`
- 3 Добавить и расширить функционал имеющихся функций.

# Что имеем?

- 1 С одной стороны, кода было сначала достаточно много



- 2 Но что работало действительно так, как нужно?

## Какие проблемы были в существующем коде

- 1 Формально кода было много, но реально работало очень мало.
- 2 Одна из основных проблем: файлы не были отделены друг от друга логически.
- 3 Как следствие, присутствовали лишние зависимости между файлами.

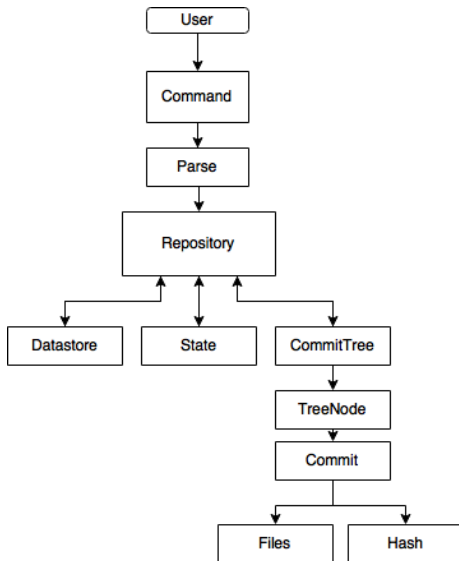
## Что происходило?

- 1 Исправление этих проблем: теперь логика объекта инкапсулирована в нем. Никто не знает о "внутренностях" других.
- 2 Добавлен `RepositoryState`, отвечающий за состояние репозитория: текущие файлы, их хеши, информация когда они добавились. Также это сериализуется вместе с репозиторием.
- 3 Добавлен `FileManager` для работы с файлами. Синхронизация с рабочей копией.
- 4 Были добавлены и отлажены команды, такие как: добавить файл(`add`), откатить его до версии последнего коммита(`revert`), откат целого коммита и создание репозитория.

## Что получилось сделать.

- 1 Были реализованы базовые команды для системы контроля версий: `init`, `add`, `revert`. Также в проекте уже почти написана команда `status`.
- 2 Попутно код почти всего проекта был переписан. Умеет консистентный вид.





## Чему я научился?

- 1 Прежде всего, перестать теряться в большом количестве файлов.
- 2 Научился разбираться в чужом коде.
- 3 Научился писать код, который нестыдно кому-то показать.
- 4 Научился работать с библиотеками `boost::filesystem`, `boost::serialization`.
- 5 Получил опыт работы с `git`. Более глубоко узнал, как это устроено изнутри.

# Что дальше?

- Система контроля версий написана с заделом на будущее. (DataStore, CommitTree, Exceptions)
- Этот код в принципе может использоваться для дальнейших, более сложных систем контроля версий.
- Дальнейше расширение функционала приложения.
- Добавление GUI. Работа с удаленным репозиторием.

# Спасибо за внимание

Репозиторий:

[https://github.com/shavkunov/au\\_git](https://github.com/shavkunov/au_git)