Отчёт по лабораторной работе 2

Первоначальная настройка git

Сидорова Наталья Андреевна

Содержание

# 1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

# 2 Задание

1. Зарегистрироваться на Github
2. Создать базовую конфигурацию для работы с git
3. Создать ключ SSH
4. Создать ключ PGP
5. Настроить подписи git
6. Создать локальный каталог для выполнения заданий по предмету

# 3 Теоретическое введение

Системы контроля версий применяются при работе нескольких человек над одним проектом. Основное дерево хранится на репозитории. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, мовмещать изменения от разных участников проекта.

# 4 Выполнение лабораторной работы

Установила git и gh (рис. [1](#fig:001)).

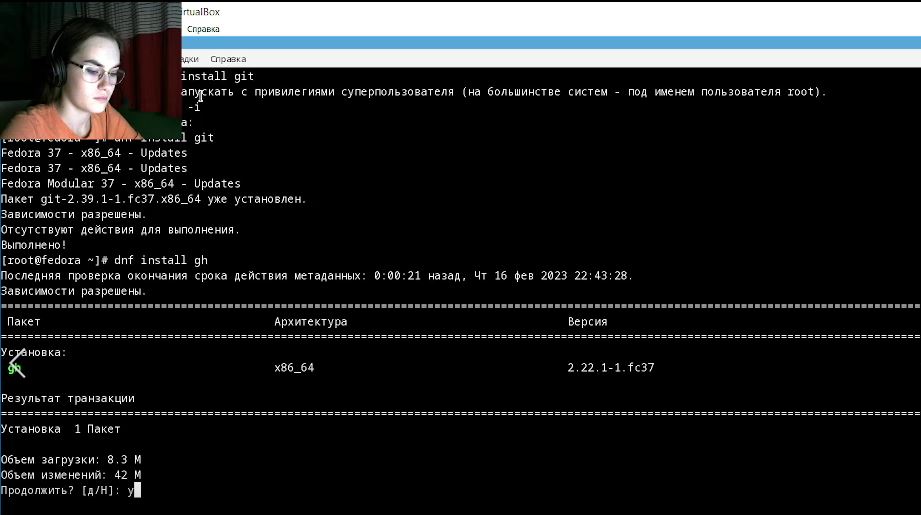


Figure 1: Установка ПО

Произвела базовую настройку git - задала имя и email владельца репозитория, настроила utf-8 в выводе сообщений git, задала имя начальной ветки(master), задала параметр autocrlf и параметр safecrlf (рис. [2](#fig:002)).

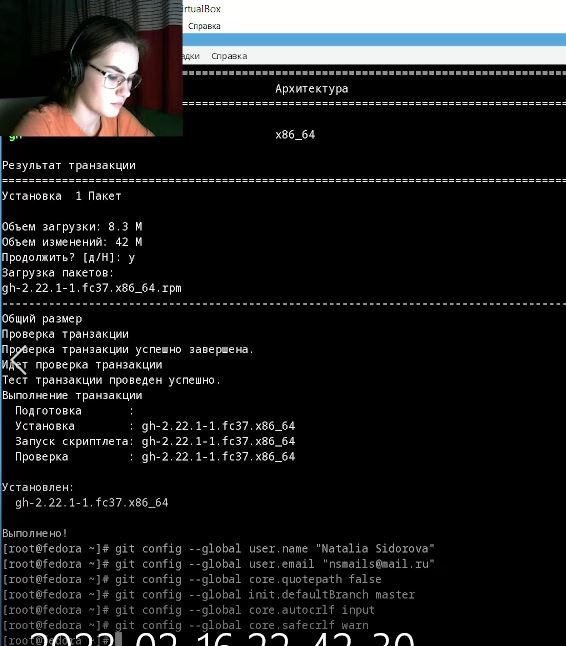


Figure 2: Базовая настройка

Создала ключи SSH - по алгоритму rsa и по алгоритму ed25519 (рис. [3](#fig:003)).

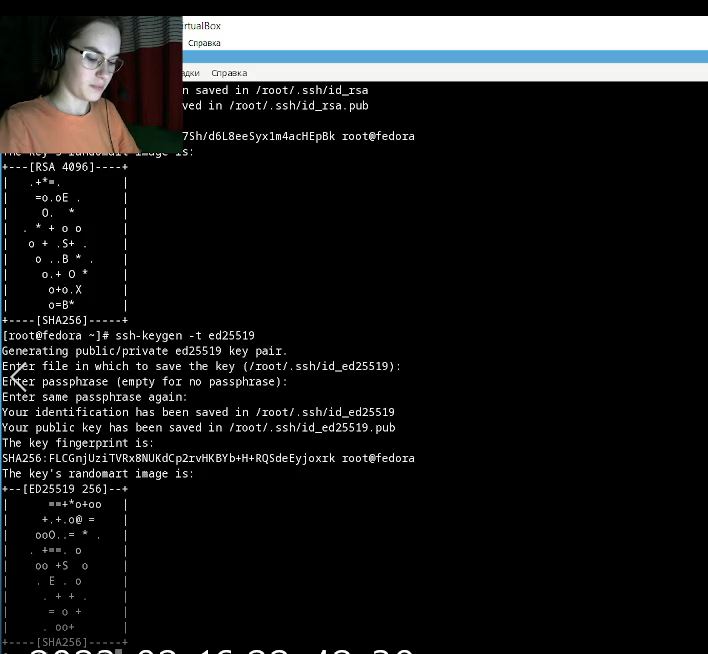


Figure 3: Ключи SSH

Создала ключ pgp (рис. [4](#fig:004)).

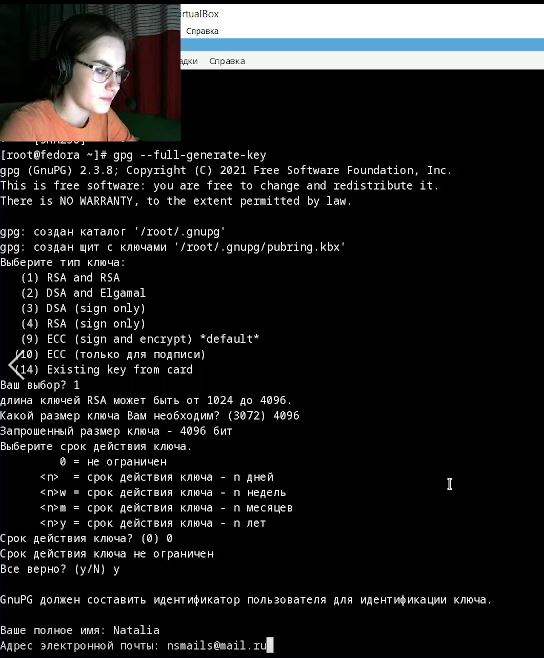


Figure 4: Ключ pgp

Учетная запись на Github уже была создана (рис. [5](#fig:005)).

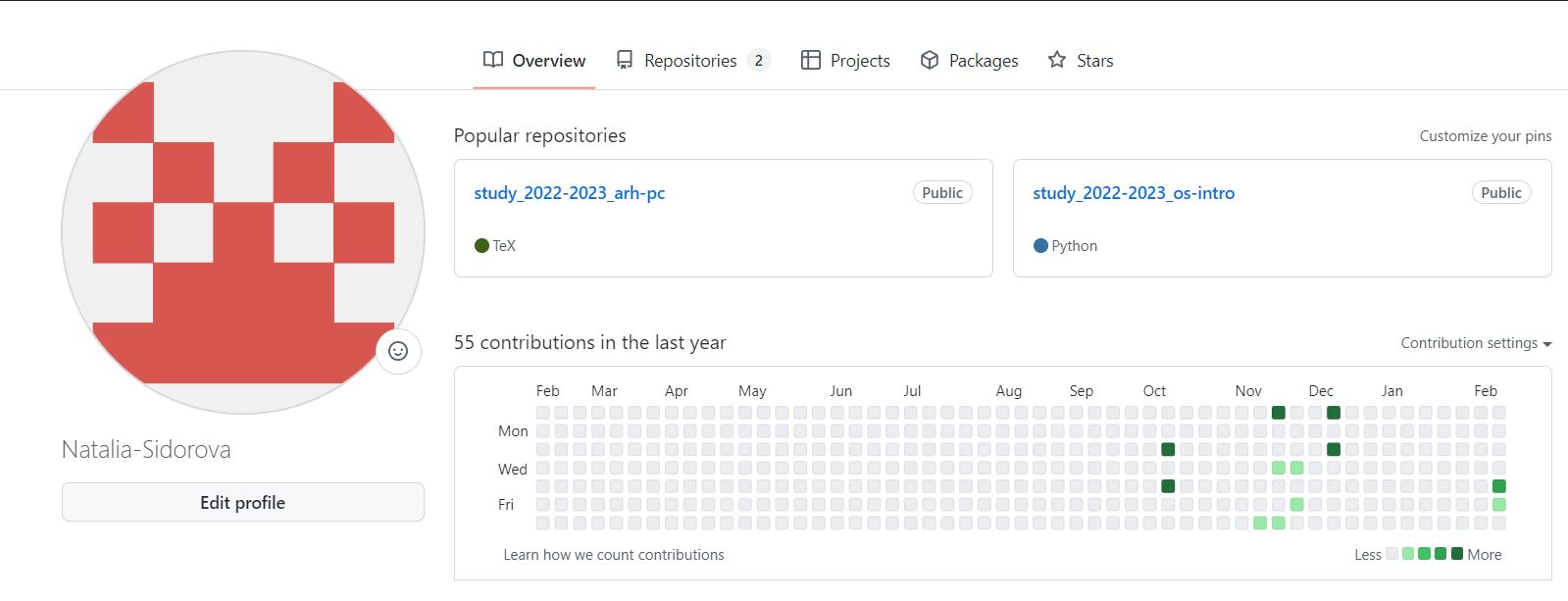


Figure 5: Учетная запись

Скопировала отпечаток приватного ключа (рис. [6](#fig:006)).

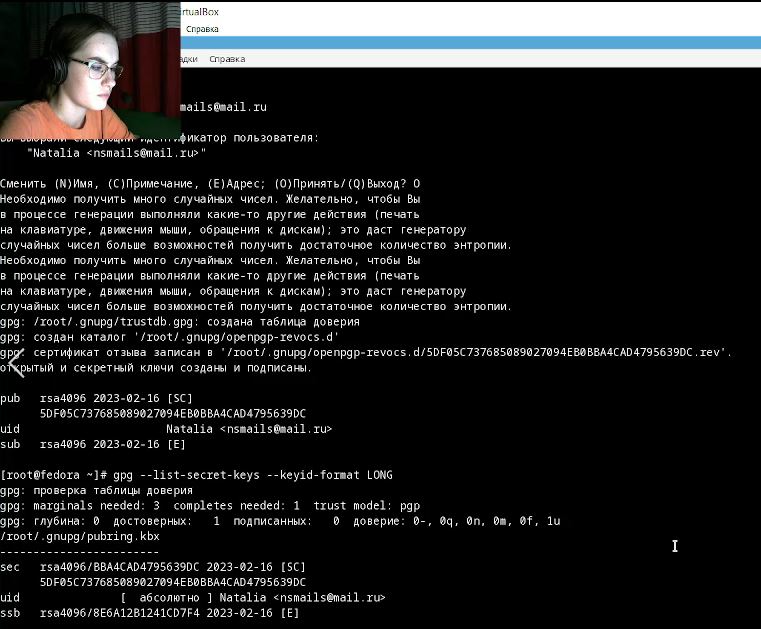


Figure 6: Отпечаток ключа

Добавила скопированный ключ в Github (рис. [7](#fig:007)).

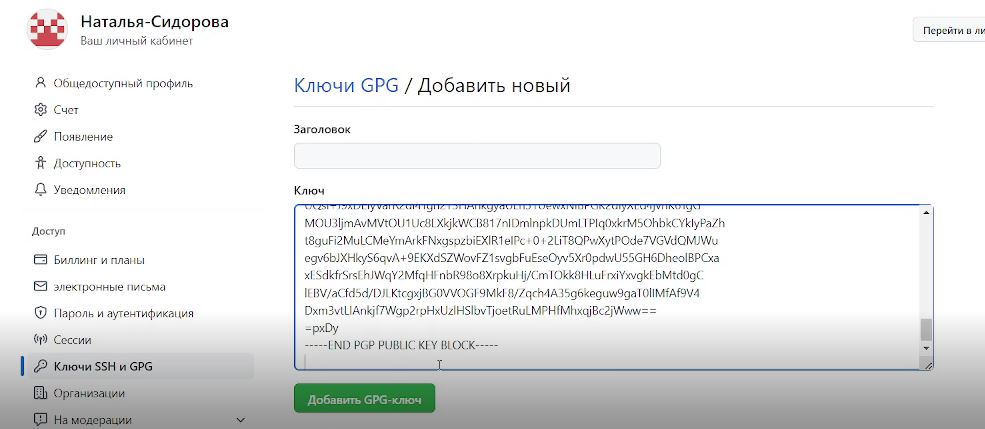


Figure 7: Добавление ключа

Настроила gh (рис. [8](#fig:008)).

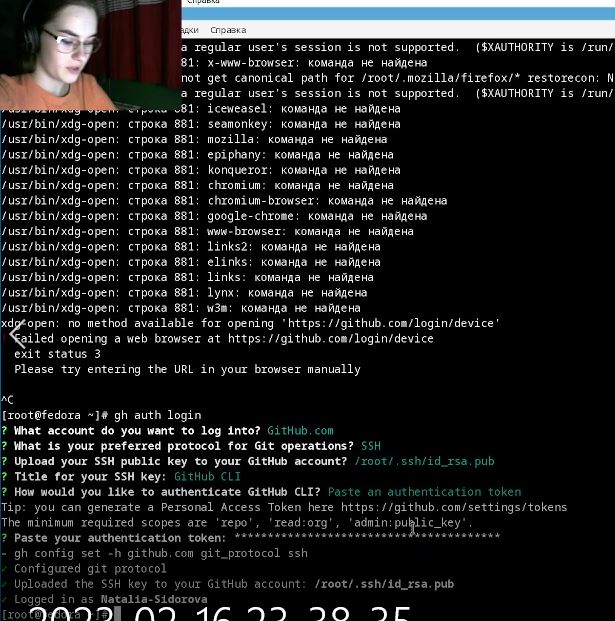


Figure 8: Настройка gh

Настроила автоматические подписи коммитов git (рис. [9](#fig:009)).



Figure 9: Автоматические подписи

Создала репозиторий курса на основе шаблона (рис. [10](#fig:010)).

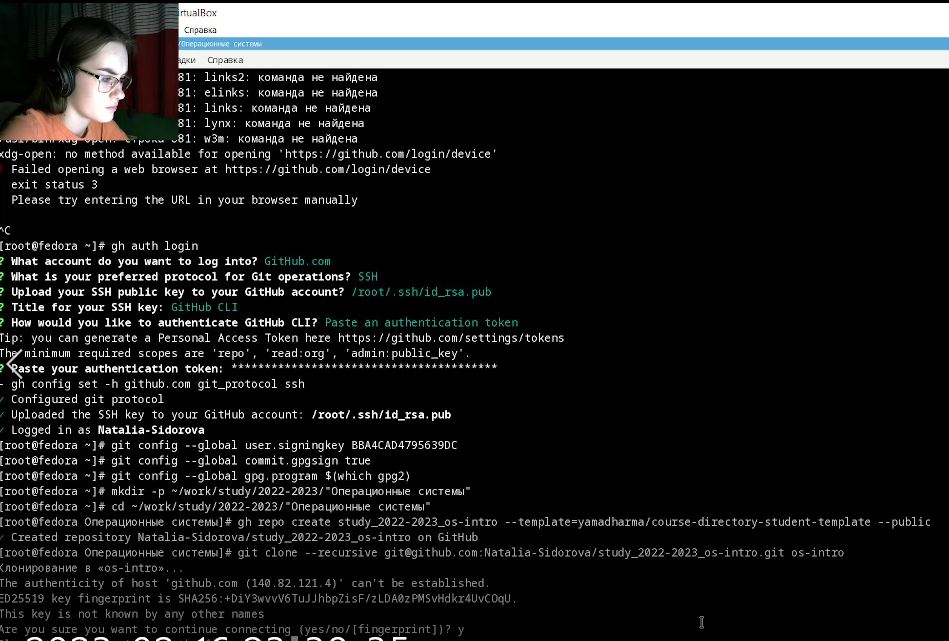


Figure 10: Репозиторий курса

Настроила каталог курса (рис. [11](#fig:011)).

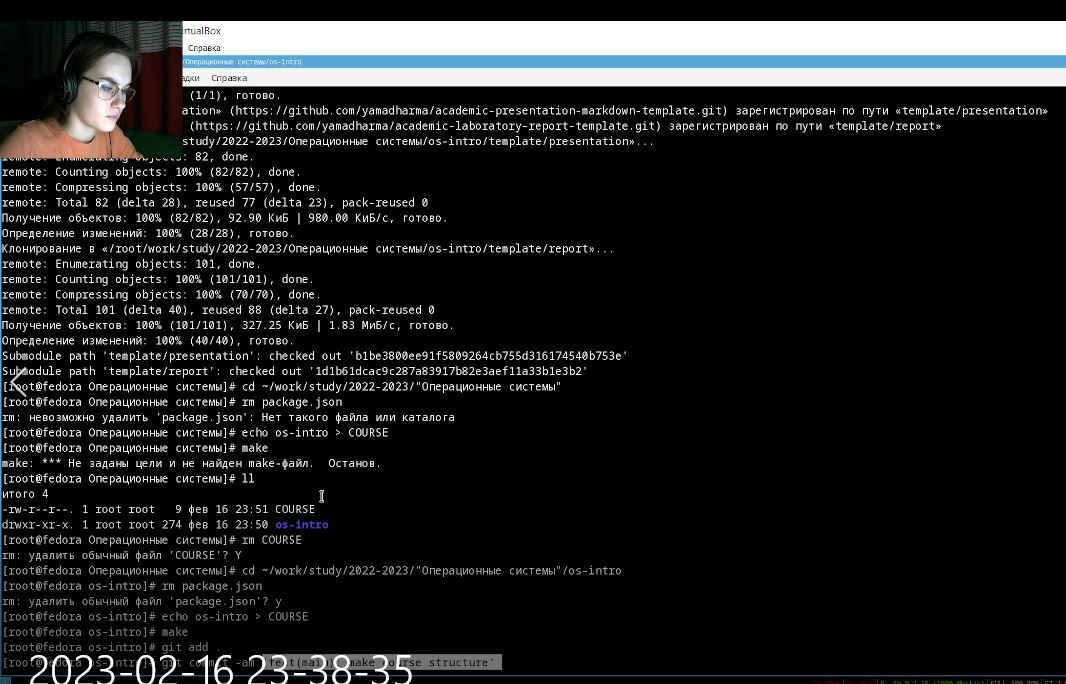


Figure 11: Настройка каталога

Отправила файлы на сервер (рис. [12](#fig:012)).

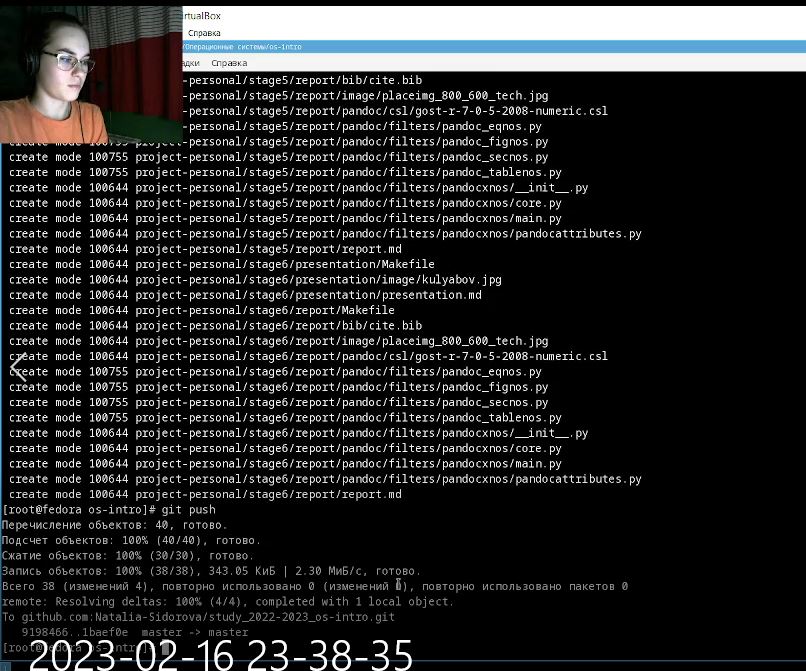


Figure 12: Отправка файлов

Ответы на контрольные вопросы:

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система контроля версий - ПО для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям. Системы контроля версий применяются для хранения полной истории изменений, причин всех производимых изменений, откат изменений, поиск причины ошибок в программе, совместная работа группы над проектом.

Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Репозиторий - хранилище версий. В нем хранятся все документы с историей их изменения, а также другая служебная информация. Сommit - отслеживание изменений, сохраняет разницу в изменениях. Рабочая копия - копия проекта, связанная с репозиторием (текущее состояние файлов проекта) История - хранит все изменения в проекте и позволяет обратиться к необходимым данным.

Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные VCS - Subversion, CVS, TFS, VAULT, AccuRev: одно основное хранилище всего проекта, каждый пользователь копирует себе необходимые файлы, изменяет их и добалвяет эти изменения обратно. Децентрализованные VCS - Git, Mercurial, Bazaar: у каждого пользователя свой вариант репозитория, есть возможность добавлять и забирать изменения из любого репозиторияБ в классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файловБ центральный репозиторий не является обязательным, выполнение большинства функций по управлению версиями осуществляется специальным сервером.

Опишите действия с VCS при единоличной работе с хранилищем. Создаем и подключаем удаленный репозиторий. Затем по мере изменения проекта отправляем изменения на сервер.

Опишите порядок работы с общим хранилищем VCS. Пользователь преред началом работы получает нужную ему версию файлов. После внесения изменений пользователь размещает новую версию в хранилище. Предыдущие версии не удаляются.

Каковы основные задачи, решаемые инструментальным средством git? Хранить информацию о всех изменениях в коде и обеспечение удобства командной работы над кодом.

Назовите и дайте краткую характеристику командам git. git init создание основого дерева репозитория git pull получение изменений текущего дерева из центрального репозитория git push - отправка всех произведенных изменений локального дерева в центральный репозиторий git status - просмотр списка измененных файлов в текущей директории git diff - просмотр текущих изменений git add. - добавить измененные файлы/каталоги git rm - удаление файла/каталога из индекса репозитория git commit -am’Описание коммита’ - сохранить все добавленные изменения и все измененные файлы git checkout -b имя ветки - создание новой ветки, базирующейся на текущей git push origin имя ветки - отправка изменений конкретной ветки в центральный репозиторий git merge -no-ff имя ветки - слияние ветки с текущим деревом git branch -d имя ветки - удаление локальной уже слитой с основным деревом ветки git branch -D имя ветки - принудительное удаления локальной ветки git push origin:имя ветки - удаление ветки с центрального репозитория

Приведите примеры использования при работе с локальным и удалённым репозиториями. git push -all(push origin master/branch)

Что такое и зачем могут быть нужны ветви (branches)? Ветвление - один из параллельных участков истории в одном хранилище, исходящих из одной версии. Обычно есть главная ветка (ствол), между ветками возможно слияние. Используются для разработки новых функций.

Как и зачем можно игнорировать некоторые файлы при commit? Во время работы над проектом могут создаваться файлы, которые не требуется добавлять в репозиторий. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore.

# 5 Выводы

В ходе проделанной работы я изучила средства контроля версий и научилась применять их, освоила работу с git, научилась подключать репозитории, добавлять и удалять файлы.

# Список литературы

1. О системе контроля версий [Электронный ресурс]. 2016. URL: https://git-scm.com/book/ru/v2/Введение-О-системе-контроля-версий .
2. Системы контроля версий [Электронный ресурс]. 2016. URL: http://uii.mpei.ru/study/courses/sdt/16/lecture02.2\_vcs.slides.pdf . ::: {#refs} :::