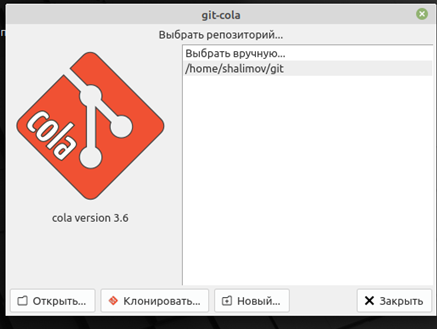
**1. Основы работы с СКВ в графическом режиме**

**Цель работы**

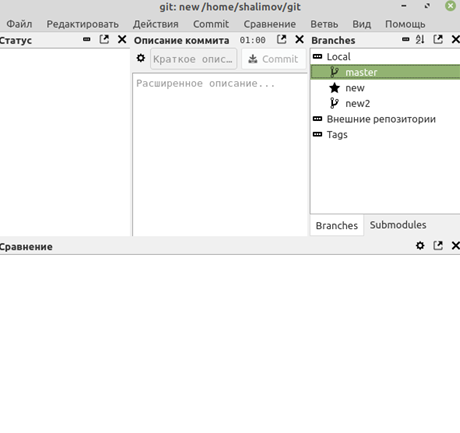
Познакомиться на практике с основными приемами работы в современных системах контроля версий.

**Задания для выполнения**

Установить на компьютер графический клиент Git.



Создайте в своей домашней папке (или в любой другой на ваш выбор) каталог, который будет содержать файлы нового программного проекта.



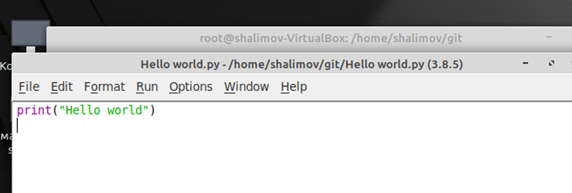
Выберите тематику программы, которую собираетесь написать. Язык программирования и используемые инструменты разработки сейчас не важны.

Инициализируйте в этой директории репозиторий гит.

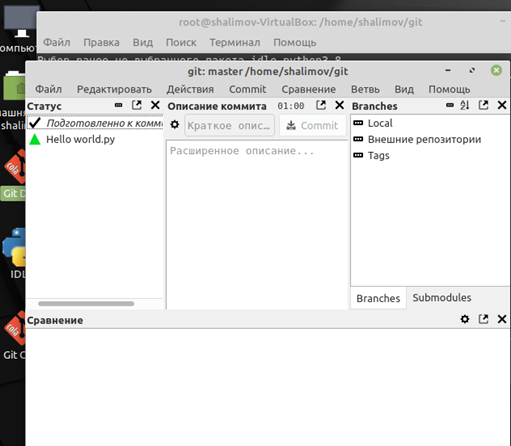
Обратите внимание на появление в этой папке скрытой подпапки с названием .git. Если вы ее не видите, то скорее всего, у вас отключено отображение скрытых папок.

Создайте новый файл для исходного текста программы. Если вы используете программный фреймворк, инициализируйте его в рабочий каталог.

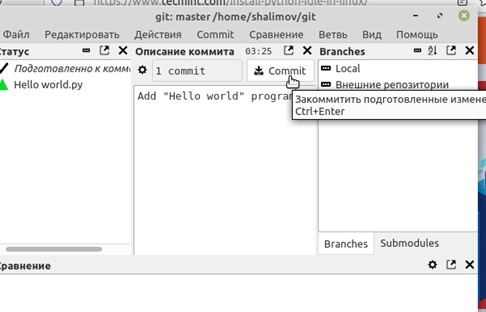
Напишите несколько строк вашей программы.



Добавьте файл с исходным текстом (несколько файлов, если необходимо) в индекс вашего репозитория.

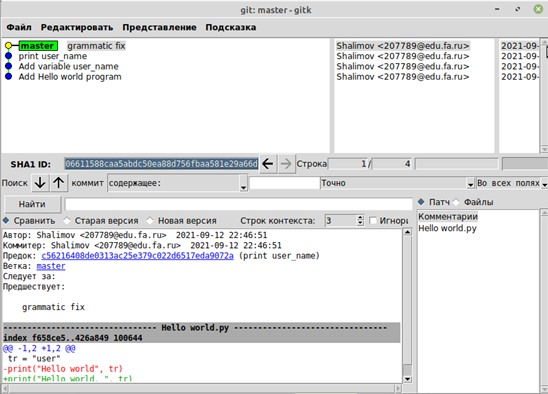


Совершите ваш первый коммит. Напишите осмысленное сообщение коммита.



Повторите несколько раз. Каждый раз, завершая определенный этап работы, выполняйте коммит и описывайте проделанные изменения в сообщении коммита.

Просмотрите историю коммитов. Попробуйте перейти на один из прошлых коммитов. Вернитесь в актуальное состояние программы.



Методические указания

Для выполнения лабораторной работы необходимо воспользоваться любым консольным клиентом системы контроля версий Git, подходящую для рабочей операционной системы. Для Windows подойдет, например, дистрибутив git-scm. Этот же пакет включает в себя и консольную утилиту git, необходимую для выполнения следующих работ. Для Linux существует, например, пакет Git Cola.

Для написания сообщений коммита необходимо придерживаться следующего общепринятого правила: в первой строке сообщения следует кратко описать произведенные изменения; если необходимо подробное описание, состоящее из многих строк, то его приводят, отступив от первой строки одну пустую. Помните, что заголовок описания коммита - это то, что будете видеть вы и ваши коллеги в истории изменений проекта.

**Контрольные вопросы**

Опишите своими словами значение следующих терминов:

рабочий каталог – каталог, версии файлы которого просматриваются/редактируются в данный момент.

репозиторий – каталоги и файлы общего проекта

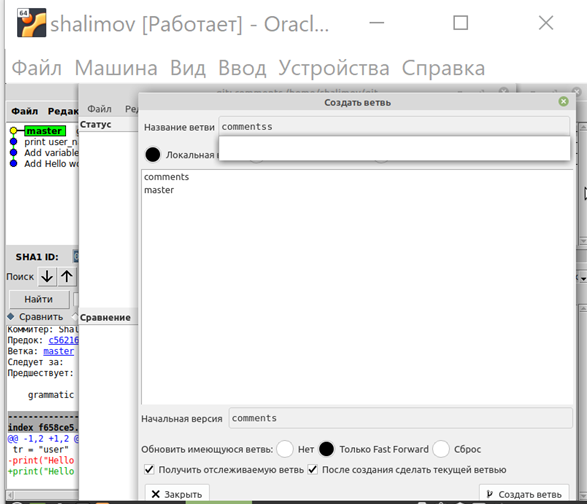
коммит – внесение накопившихся изменений в содержимом репозитория в память репозитория (сохранение)

ветка – последовательная история коммитов одного репозитория.

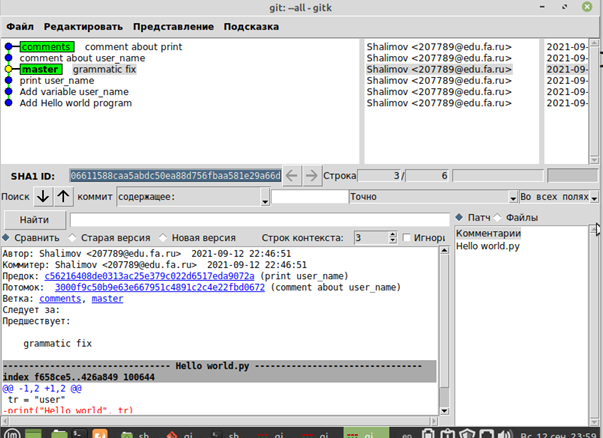
Ознакомьтесь с гайдом по выбранной вами программе-клиенту Git.

**Дополнительные задания**

Представьте, что вы начинаете большой раздел работы. Для изоляции изменений создайте новую ветку. Назовите ее, чтобы было понятно, что вы в ней будете делать.



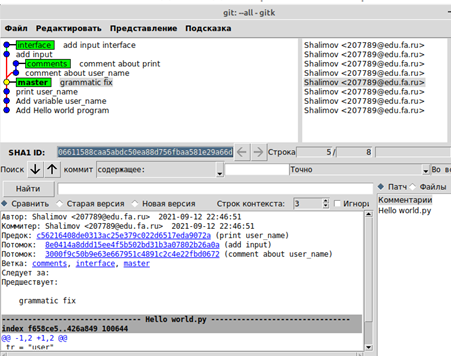
Перейдите в новую ветку и сделайте несколько коммитов.



Перейдите в основную ветку и обратите внимание на состояние рабочей директории.

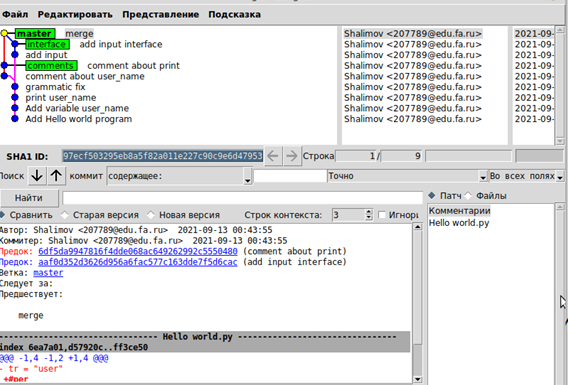
Создайте еще одну ветку для работы над другим направлением в вашей программе. Обычно так работают в команде, каждый участник в собственной ветке. Либо в ветках может идти параллельная работа над разными возможностями программы. В таком случае эти ветки называются тематическими.

Сделайте несколько коммитов во вновь созданную ветку.



Перейдите в основную ветку и слейте в нее первую тематическую ветку.

Слейте в основную ветку вторую тематическую. Если возникли конфликты слияния, разрешите их и завершите слияние.



Удалите более не нужные тематические ветки. Обратите внимание в истории, что даже при удалении веток никакие коммиты не теряются.

