МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Отчет по лабораторной работе №1

на тему «Система контроля версий»

по дисциплине «Конструирование программного обеспечения»

Кафедра программного обеспечения информационных систем и технологий

Вариант 7

Выполнил

студент группы 10701122 Сивец Д.И.

Проверил ст.пр. Станкевич С.Н.

Минск 2024

**Цель работы:** ознакомиться с системами контроля версий для разработки программного обеспечения. Освоить способы организации работы в системах контроля версий. Получить навыки командной работы.

**Постановка задачи:** выполните вышеуказанные упражнения, покажите результаты и ход выполнения укажите в отчете подтверждая скриншотами. Выберите приемлемую вам систему контроля версий, зарегистрируйтесь в ней. Создайте команду разработчиков, добавив членов команды. Создайте проект (предприятие), и начните работу. Название проекта должно содержать атрибуты нашей дисциплины, номера лабораторной работы, имена разработчиков и т.п. Ваш проект должен легко идентифицироваться среди тысяч других. Желательно ограничить доступ к проекту, но у преподавателя должна быть возможность просматривать и работать с этим проектом. Напишите отчет. Ключевые моменты вашей работы должны быть отражены с помощью скриншотов. Отчет должен храниться в проекте в системе контроля версий.

**Выполнение работы:**

Создание нового репозитория:

Рисунок 1 — инициализация репозитория

После заполнения проекта кодом, он выглядит следующим образом:

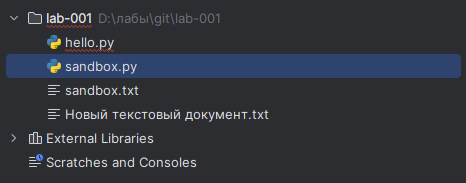
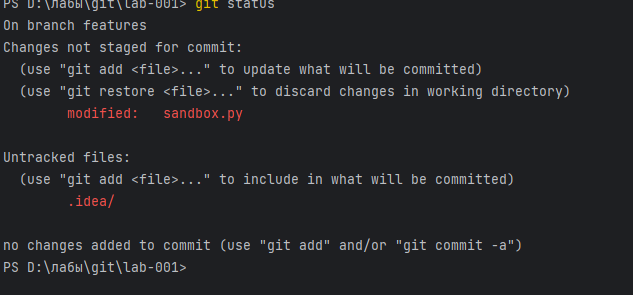


Рисунок 2 — структура репозитория

Чтобы проиндексировать файлы (из unstaged сделать staged) используем команду git add:

Рисунок 3 — git status до git add

Рисунок 4 — git status после git add

Для сохранения изменений в репозиторий используется git commit.

Рисунок 4 — git commit

Для улучшения проекта будет использоваться ветка features. Чтобы ее создать и сразу переключиться на нее используем следующую команду:



Рисунок 5 — git checkout

**Вывод:** в результате выполнения лабораторной работы было выполнено знакомство с VCS для разработки программного обеспечения.

**Индивидуальное задание:**

Лиценизия MIT — лицензия свободного ПО, разработанная Массачусетским технологическим институтом. Данная лицензия разрешает использовать код при разработке закрытого ПО с указанием авторства.

Текст лицензии выглядит следующим образом:

MIT License

Copyright (c) 2023 Schedule makers

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE

**Контрольные вопросы:**

**1. Что такое системы контроля версий?**

Система контроля версий (VCS) — это программное обеспечение, которое помогает разработчикам отслеживать изменения в файловой системе и эффективно управлять версиями файлов и кода в проекте.

**2. В каких сферах деятельности могут использоваться системы контроля версий?**

Программирование, дизайн и графика, научные исследования.

**3. Назовите наиболее популярные СКВ для разработки ПО.**

Наиболее популярные СКВ это Git, Mercurial, Subversion.

**4. Какую СКВ вы выбрали для работы и почему?**

**5. Какие виды и типы контроля версий существуют?**

Существуют два основных типа систем контроля версий: централизованные (CVCS) и распределённые (DVCS).

**6. Что такое GIT?**

Git — это система контроля версий, которая помогает отслеживать историю изменений в файлах.

**7. Знаете ли вы историю создания GIT?**

Нет, не знаю.

**8. Кто является создателем GIT?**

Git был создан Линусом Торвальдсом в 2005 году.

**9. Как слово «git» переводится с английского языка?**

Слово “git” в переводе с английского может означать "негодяй".

**10.Укажите основные команды, компоненты и процедуры работы в СКВ.**

Основные команды Git включают git init (создание нового репозитория), git add (добавление файлов в индекс), git commit (сохранение изменений), git branch (создание новой ветки), git merge (слияние веток), и многие другие.

**11. Что такое система управления версиями?**

Система управления версиями — это программное обеспечение, которое помогает отслеживать изменения в коде с течением времени.

**14. Как создать ветку?**

Для создания ветки в Git, вы можете использовать команду git branch <имя ветки>.

**15. Как провести слияние? Как разрешить конфликт и что это такое?**

Для слияния веток в Git, вы можете использовать команду git merge <имя ветки>. Если возникает конфликт, вы можете его разрешить, отредактировав файлы вручную и затем добавив их с помощью git add.

**16. Как зафиксировать изменения?**

Для фиксации изменений в Git, вы можете использовать команду git commit -m "<сообщение>".

**17. Как провести откат? Различия в reset и revert, мягкий и жесткий reset.**

Для отката изменений в Git, вы можете использовать команды git reset или git revert. git reset перемещает указатель HEAD и, опционально, изменяет индекс или рабочую директорию, в то время как git revert создает новый коммит, который отменяет изменения, внесенные в указанный коммит.

**18. Какова последовательность действий при работе с локальным репозиторием?**

При работе с локальным репозиторием, вы можете начать с git init, затем использовать git add для добавления файлов, git commit для сохранения изменений, и так далее.

**19. Какова последовательность действий при работе с удаленным репозиторием?**

При работе с удаленным репозиторием, вы можете начать с git clone <url>, затем использовать git pull для получения последних изменений, git push для отправки своих изменений на сервер, и так далее.

**20. Каковы возможности при работе с удаленным репозиторием? Как его клонировать, получать и отправлять данные?**

При работе с удаленным репозиторием, вы можете клонировать его с помощью git clone <url>, получать данные с помощью git pull, и отправлять данные с помощью git push.

**21. Что будет если в процессе работы изменить название папки с рабочим проектом, у нас это «песочница».**

Если вы измените название папки с рабочим проектом, Git все равно будет отслеживать файлы внутри этой папки, так как он отслеживает содержимое файлов, а не их пути. Однако, если вы переместите папку вне рабочего каталога Git, Git больше не будет отслеживать эти файлы.