Министерство образования Республики Беларусь Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования Кафедра инженерной психологии и эргономики

# ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ ИНФОРМАЦИОННЫХ СИСТЕМ

ОТЧЕТ

по практическому занятию №1

# «КУРС ОБУЧЕНИЯ GIT НА РУССКОМ»

Выполнил: ст. гр. 210902 Глушко А. А.

Проверил: Давыдович К.И.

Минск 2024

**Настройка пользователя**

git config --global user.name "Artem"

* + Устанавливает имя пользователя для всех репозиториев Git на компьютере. Это имя будет отображаться в историях коммитов.

git config --global user.email "GIT44@gmail.com"

* + Устанавливает электронную почту пользователя для всех репозиториев Git. Она также будет видна в коммитах.

**Настройка глобальных параметров Git**

git config --global init.defaultBranch main

* + Устанавливает имя основной ветки (по умолчанию) в новых репозиториях как main, что является современным стандартом.

git config --global core.autocrlf true

* + Включает автоматическую конвертацию символов конца строки, чтобы избежать проблем с переносами строк между Windows и Unix системами.

git config --global core.safecrlf warn

* + Настройка, которая предупреждает о смешанных окончаниях строк в файлах. Если такие проблемы будут обнаружены, Git выдаст предупреждение.

**Создание рабочей директории и файла**

mkdir work

* + Создает новую директорию с именем work.

cd work

* + Переходит в созданную директорию work.

New-Item -Path . -Name "hello.html" -ItemType "file"

* + Создает новый файл с именем hello.html в текущей директории (используется в PowerShell).

Инициализация Git-репозитория

git init

* + Инициализирует новый Git-репозиторий в текущей директории, создавая скрытую папку .git.

**Добавление и коммит файла**

git add hello.html

* + Добавляет файл hello.html в индекс (staging area), подготавливая его к коммиту.

git commit -m "Initial Commit"

* + Создает коммит с сообщением "Initial Commit", фиксируя текущее состояние файлов в репозитории.

Проверка статуса репозитория

git status

* + Показывает текущее состояние рабочей директории и индекса, включая изменения, которые еще не были закоммичены.

**Работа с файлами и ветками**

git add

* + Команда не завершена, но обычно используется для добавления изменений в индекс.

git checkout

* + В зависимости от контекста, используется для переключения между ветками или для восстановления файлов из другой ветки.

**Добавление и коммит изменений**

git add a.html git add b.html

* + Добавляет файлы a.html и b.html в индекс.

git commit -m "Changes for a and b"

* + Создает коммит с сообщением "Changes for a and b", фиксируя изменения в файлах a.html и b.html.

git add c.html

* + Добавляет файл c.html в индекс.

git commit -m "Unrelated change to c"

* + Создает коммит с сообщением "Unrelated change to c", фиксируя изменения в файле c.html.

**Редактирование коммита**

git commit Added h1 tag

* + Не завершенная команда; предполагалось добавить изменения с сообщением о добавлении тега h1. Должно быть записано с флагом -m.

Esc :wq enter

* + Команды для выхода из текстового редактора Vim. Esc переводит в командный режим, :wq сохраняет изменения и выходит.

**Подготовка к коммиту всех изменений**

git add .

* + Добавляет все изменения в текущей директории и поддиректориях в индекс.

**Просмотр истории коммитов**

git log

* + Показывает историю коммитов в репозитории.

git log --pretty=oneline

* + Показывает историю коммитов в компактном формате, где каждый коммит отображается в одной строке.

git log --all --pretty=format:"%h %cd %s (%an)" --since="7 days ago"

* + Показывает все коммиты за последние 7 дней с указанием хеша, даты, сообщения и имени автора.

git log --pretty=format:"%h %ad | %s%d [%an]" --date=short

* + Выводит историю коммитов в формате, который включает хеш, дату, сообщение, метки и имя автора, причем дата отображается в коротком формате.

--pretty="..." — определяет формат вывода.

%h — укороченный хеш коммита.

%ad — дата коммита.

| — просто визуальный разделитель.

%s — комментарий.

%d — дополнения коммита («головы» веток или теги).

%an — имя автора.

--date=short — сохраняет формат даты коротким и симпатичным.

Сохранение формата лога

git config --global format.pretty '%h %ad | %s%d [%an]' git config --global log.date short

10

git checkout <hash> меняет текущую точку cat hello.html показывает содержимое

git switch main вернуться к последней точке в main

11

git tag v1 тег для текущей версии

git checkout v1^ Переключение на предшествующий коммит git tag просмотр тегов

13

git reset HEAD hello.html сбрасывает текущие add

14

git revert HEAD отменяет последний коммит либо коммит по хешу

15

git reset --hard v1 сброс к предшествующему коммиту git log --all показывает все коммиты

16

git tag -d oops удаление тега

17

git commit --amend -m "Added copyright statement with email" изменение предыдущего коммита

18

git switch -c style создание новой ветки

19 git switch main переключение между ветками

20

git log hello.html просмотр изменений в конкретном файле git log style.css

git show v1 просмотр изменений в коммите

mv hello.html index.html переименование файла git add . добавление в коммит

mkdir css

git mv style.css css/style.css перемещение файла безопасным способом git log --follow css/style.css лог файла с историей до переименования

21

git add README.txt добавление файла в ветку

22

git log --all --graph просмотр веток и их расхождений

23

git switch style

git merge main слияние

25

git merge --abort отмена слияния

27

git reset --hard HEAD~2 сброс до коммита который находится за два коммита до head

30

git clone work home клонирование репозитория ls просмотр репозитория

31

git remote получение имени удаленного репозитория git remote show origin более подробная информация

32

git branch просмотр веток доступных в клонированном репозитории git branch -a список удаленных веток

35

git fetch подтягивание коммитов из удаленного репозитория

37

git branch --track style origin/style добавление локальной ветки, которая отслеживает локальную ветку

38

git clone --bare work work.git создание чистого репозитория

39

git remote add shared ../work.git добавление чистого репозитория как удаленного к оригинальному репозиторию

40

git push shared main отправить изменения в общий репозиторий