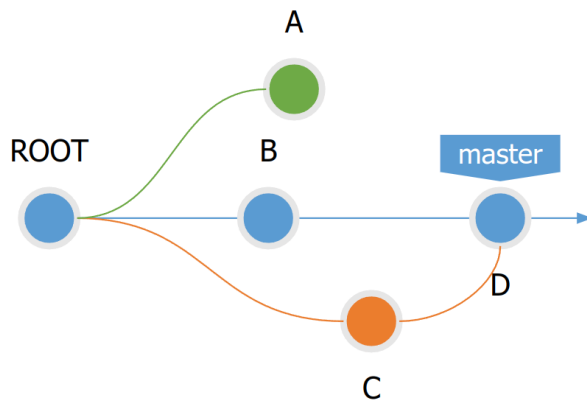


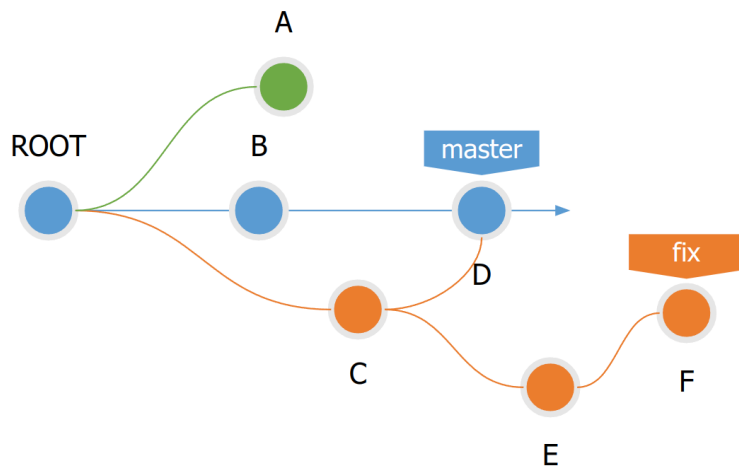
Вопросы по Git

1. Какой командой можно инициализировать новый локальный репозиторий?
 - `git init`
 - `git clone`
 - `git pull`
2. Какой командой можно посмотреть текущий статус репозитория?
 - `git status`
 - `git show`
 - `git log`
3. Какой командой добавляются изменения в индекс для последующего коммита?
 - `git add`
 - `git push`
 - `git branch`
4. Какой командой можно создать новый коммит?
 - `git commit`
 - `git merge`
 - `git pull`
5. Какой командой клонируется удалённый репозиторий на локальный компьютер?
 - `git clone`
 - `git init`
 - `git commit`
6. Какой командой можно увидеть историю коммитов?
 - `git log`
 - `git status`
 - `git diff`
7. Какой командой можно создать новую ветку?
 - `git branch`
 - `git add`
 - `git push`
8. Какая команда отображает изменения в рабочем каталоге?
 - `git diff`
 - `git log`
 - `git fetch`
9. Какой командой можно увидеть, в какой ветке вы находитесь?
 - `git branch`
 - `git add`
 - `git push`
10. Какой командой можно переместиться в другую ветку?
 - `git checkout <branch_name>`
 - `git init <branch_name>`
 - `git reset <branch_name>`

11. Расставьте по приоритету (от высшего к низшему) настройки Git на следующих уровнях:
- Системные
 - Глобальные
 - Локальные
12. Расставьте команды в правильном порядке для создания первого в новом репозитории (предполагается, что настройки имени пользователя и email уже выполнены, а инициализация репозитория – нет):
- Создание, добавление, редактирование файлов
 - `git add .`
 - `git init`
 - `git commit -m 'сообщение коммита'`
13. Для указания правил, определяющих игнорируемые файлы Git, используется по умолчанию файл с именем _____
14. Режим, при котором указатель HEAD указывает на конкретный коммит, а не на ветку, называется _____. При создании коммитов в этом режиме они автоматически становятся dangling commit и через какое-то время будут удалены в рамках GC (при условии, что на них не будут ссылаться никакие другие указатели).
Варианты ответов:
- Direct HEAD
 - Detached HEAD
 - Commit HEAD
15. Какие коммиты относятся к ветке master?



- ROOT
 - A
 - B
 - C
 - D
16. Какие коммиты являются висящими?



- ROOT
- A
- B
- C
- D
- E

17. Команда `git reset --hard`:

- Оставляет текущие изменения в рабочем каталоге и индексе
- Не оставляет текущие изменения ни в рабочем каталоге, ни в индексе
- Оставляет текущие изменения только в рабочем каталоге

18. Команда `git reset --mixed`:

- Оставляет текущие изменения в рабочем каталоге и индексе
- Не оставляет текущие изменения ни в рабочем каталоге, ни в индексе
- Оставляет текущие изменения только в рабочем каталоге

19. Команда `git reset --soft`:

- Оставляет текущие изменения в рабочем каталоге и индексе
- Не оставляет текущие изменения ни в рабочем каталоге, ни в индексе
- Оставляет текущие изменения только в рабочем каталоге

20. С помощью какой команды можно посмотреть diff между рабочим каталогом и индексом:

- `git diff`
- `git diff --cached`
- `git diff HEAD`

21. С помощью какой команды можно посмотреть diff между рабочим каталогом и текущим коммитом:

- `git diff`
- `git diff --cached`
- `git diff HEAD`

22. С помощью какой команды можно посмотреть diff между индексом и текущим коммитом:

- `git diff`
- `git diff --cached`
- `git diff HEAD`

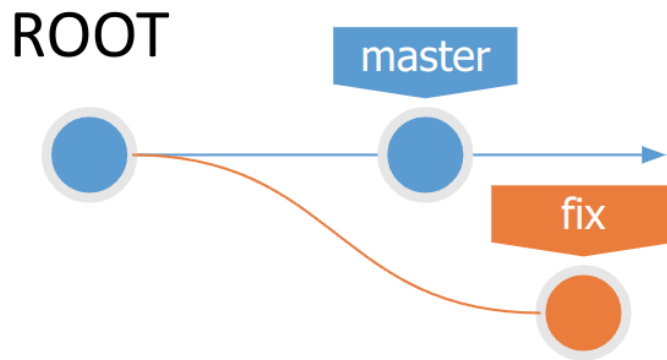
23. Для слияния двух и более веток используется команда:

- `git checkout`
- `git merge`
- `git reset`
- `git commit`

24. Выберите правильный порядок разрешения конфликта при слиянии:

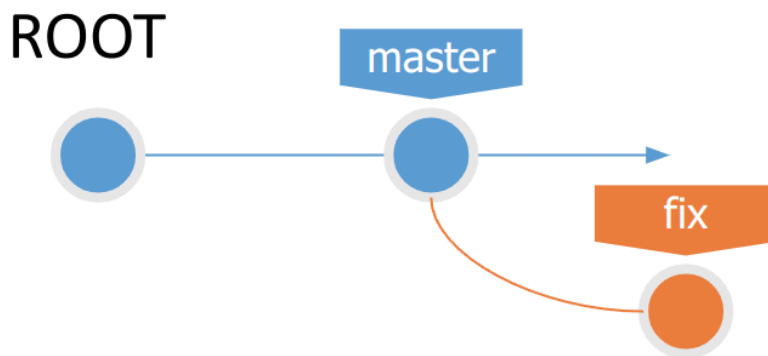
- `git add .`
- Удаление маркеров из файлов с конфликтами
- `git commit`

25. Получится ли смёржить ветку `fix` в ветку `master` с использованием Fast-Forward:



- Да
- Нет

26. Получится ли смёржить ветку `fix` в ветку `master` с использованием Fast-Forward:



- Да
- Нет

27. С помощью какой команды можно отправить изменения из локального репозитория в удалённый:
- `git commit`
 - `git push`
 - `git fetch`
 - `git pull`
28. С помощью какой команды (или команд) можно получить изменения из удалённого репозитория в локальный:
- `git commit`
 - `git push`
 - `git fetch`
 - `git pull`
29. При использовании какой команды осуществляется попытка автоматически выполнить merge:
- `git fetch`
 - `git pull`
30. Выберите инструменты для удаления "секретов" и других чувствительных данных из репозитория:
- Trufflehog
 - `git filter-repo`
 - BFG Repo-Cleaner
31. Какая команда позволяет применить изменения из одного коммита на текущую ветку, не создавая новый коммит, а сохраняя авторство и сообщение исходного коммита?
- `git cherry-pick`
 - `git merge --squash`
 - `git rebase`
32. Какой командой можно временно сохранить изменения в локальном рабочем каталоге, не добавляя их в коммит, и вернуться к ним позже?
- `git stash`
 - `git reset --soft`
 - `git clean`
33. Какая команда позволяет повторить изменения из одного коммита и применить их поверх текущей ветки как новый коммит?
- `git cherry-pick`
 - `git rebase -i`
 - `git amend`
34. При выполнении команды `git rebase -i` в интерактивном режиме, какой параметр используется для объединения нескольких коммитов в один?
- `squash`
 - `drop`
 - `edit`
35. Какая команда удаляет все неотслеживаемые файлы и директории из рабочего каталога?

- `git clean -fd`
 - `git reset --hard`
 - `git rm -rf`
36. Какая команда позволяет изменить только сообщение последнего коммита без создания нового?
- `git commit --amend`
 - `git rebase -i HEAD~1`
 - `git reset --soft HEAD^`
37. В каких случаях нужно выполнять команду `git rebase`, а не `git merge`?
- Когда требуется сохранить линейную историю без создания дополнительных коммитов слияния
 - Когда нужно объединить ветки с разными историями изменений
 - Когда нужно откатить последние изменения на ветке
38. Какая команда позволяет перенести все незакоммиченные изменения в другой коммит?
- `git stash save`
 - `git stash pop`
 - `git stash branch`
39. Какой параметр `git log` используется для отображения всех коммитов, которые есть в одной ветке, но отсутствуют в другой?
- `git log branch1 ^branch2`
 - `git log branch1 --branch2`
 - `git log --left-right branch1...branch2`
40. Какой параметр используется в `git diff`, чтобы показать изменения в коммитах только для конкретного файла?
- `git diff <commit1> <commit2> -- <file>`
 - `git diff <commit1> -- <commit2> --file`