**Методические указания**

**по выполнению практической работы № 1: «Системы контроля версий»**

по дисциплине «Технологии разработки программных приложений»

**Часть 1 (1 неделя). Основные команды git**

Пройдите курс на сайте<https://githowto.com/> (до раздела 22). Обязательно должны быть выполнены следующие действия:

1. Установите и настройте клиент git на своей рабочей станции.

2. Создайте локальный репозиторий и добавьте в него несколько файлов.

3. Внесите изменения в один из файлов.

4. Проиндексируйте изменения и проверьте состояние.

5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.

7. Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

8. Верните рабочий каталог к одному из предыдущих состояний.

9. Изучите, как создавать теги для коммитов для использования в будущем.

10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

11. Отмените один из коммитов в локальном репозитории.

**Часть 2 (2 неделя). Ветвление и разрешение конфликтов**

Продолжите проходить курс на сайте<https://githowto.com/> (начиная с раздела 23) или курс на сайте<https://learngitbranching.js.org/?locale=ru_RU>.

**Часть 3 (3 неделя). Системы управления репозиториями**

1. Создайте аккаунт на GitHub.

2. Создайте репозиторий на GitHub

3. Создайте новый локальный репозиторий с несколькими файлами на рабочей станции и загрузите его содержимое на GitHub.

4. Чтобы избежать ввода логина и пароля, создайте SSH-ключ для авторизации.

5. Создайте в репозитории новую ветку. Произведите в ней несколько изменений и слейте с веткой master.

6. Выполните цепочку действий в репозитории, согласно варианту. Вариант совпадает с <остатком от деления (номера студента в списке -1) на 15+ 1> (16-й студент выполняет задание 1 варианта).

|  |  |
| --- | --- |
| № варианта | Задание |
| 1 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке  4) Выгрузите изменения в удаленный репозиторий  5) Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории)  6) Выведите в консоли различия между веткой master и новой веткой  7) Слейте новую ветку с master при помощи merge |
| 2 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите коммит в ветке master  4) Произведите 3 коммита в новой ветке в разные файлы  5) Выгрузите изменения в удаленный репозиторий  6) Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории)  7) Выведите в консоли различия между веткой master и новой веткой  8) Перебазируйте новую ветку на master |
| 3 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте тег указывающий на последний коммит в ветке master  3) Создайте новую ветку и выведите список всех веток  4) Произведите 3 коммита в новой ветке  5) Выгрузите все изменения в удаленный репозиторий  6) Откатите ветку к созданному тегу (в том числе в удаленном репозитории)  7) Выведите в консоли различия между веткой master и новой веткой |
| 4 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Откатите в новой ветке предпоследний коммит (в том числе в удаленном репозитории)  6) Выведите в консоли различия между веткой master и новой веткой  7) Слейте новую ветку с master при помощи merge |
| 5 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Произведите revert предпоследнего коммита в новой ветке  6) Выведите в консоли различия между веткой master и новой веткой  7) Слейте новую ветку с master при помощи merge |
| 6 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в новых файлах  4) Выгрузите изменения в удаленный репозиторий  5) Произведите какие-нибудь изменения в файле который существует в ветке master, но не коммитьте их  6) Выполните git stash  7) Переключитесь на ветку master  8) Выполните git stash pop  9) Выведите в консоли различия между веткой master и новой веткой  10) Слейте новую ветку с master при помощи merge |
| 7 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте две новых ветки и выведите список всех веток  3) Произведите 3 коммита в первой ветке  4) Переместите (cherry pick) первый коммит во вторую ветку  5) Выгрузите изменения в удаленный репозиторий  6) Выведите в консоли различия между веткой master и новыми ветками  7) Слейте обе ветки с master при помощи merge |
| 8 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте две новых ветки и выведите список всех веток  3) Произведите по 2 коммита в обоих ветках в новых файлах  4) Выгрузите изменения в удаленный репозиторий  6) Выведите в консоли различия между веткой master и обоими ветками  7) Слейте одну ветку с master при помощи merge  8) Подлейте master ко второй ветке  9) Выведите в консоли различия между веткой master и обоими ветками |
| 9 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 5 коммитов в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Произведите revert 2-го и 4-го коммита в новой ветке  6) Выведите в консоли различия между веткой master и новой веткой  7) Слейте новую ветку с master при помощи merge |
| 10 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Произведите какие-нибудь изменения в файле который существует в ветке, но не коммитьте их  6) Внесите эти изменения в последний коммит (amend)  7) Выведите в консоли различия между веткой master и новой веткой  8) Слейте новую ветку с master при помощи merge |
| 11 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Добейтесь того, чтобы эти три изменения были представлены одним коммитом  6) Выведите в консоли различия между веткой master и новой веткой  7) Слейте новую ветку с master при помощи merge |
| 12 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке в разные файлы  4) Выгрузите изменения в удаленный репозиторий  5) Произведите какие-нибудь изменения в файле, который существует в ветке, но не коммитьте их  6) Выведите в консоли различия между новой веткой и рабочей директорией  7) При помощи команд git приведите файл из п.5 к состоянию до изменения  8) Слейте новую ветку с master при помощи merge |
| 13 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите 3 коммита в новой ветке с одним и тем же файлов  4) Выгрузите изменения в удаленный репозиторий  5) Выведите в консоли информацию о том, в каком коммите была изменена каждая строка файла  6) Слейте новую ветку с master при помощи создания pull request-а |
| 14 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Создайте новый файл в репозитории, но не коммитьте его  3) Произведите 3 коммита в новой ветке в другие файлы, не добавляя данный файл в git  4) Выведите в консоли результат git status  5) Добавьте файл из п.3 в .gitignore и закоммитьте это в репозиторий  6) Выгрузите изменения в удаленный репозиторий  7) Выведите в консоли различия между веткой master и новой веткой  8) Слейте новую ветку с master при помощи merge |
| 15 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Создайте новый файл в репозитории, но не коммитьте его  3) Произведите 3 коммита в новой ветке в другие файлы, не добавляя данный файл в git  4) Выведите в консоли результат git status  5) При помощи команд git очистите рабочую директорию от untracked файла  6) Выгрузите изменения в удаленный репозиторий  7) Выведите в консоли различия между веткой master и новой веткой  8) Слейте новую ветку с master при помощи merge |

7. Выполните общее задание:

1. Сделайте форк репозитория в соответсвии с вариантом (вариант является <остатком от деления (номера студента по списку -1) на 10 + 1>
2. Склонируйте его на локальную машину
3. Создайте две ветки branch1 и branch2 от последнего коммита в master'е
4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла
5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом
6. Выгрузите все изменения во всех ветках в удаленный репозиторий
7. Проведите еще 3 коммита в ветку branch1
8. Склонируйте репозиторий еще раз в другую директорию
9. В новом клоне репозитории сделайте 3 коммита в ветку branch1
10. Выгрузите все изменения из нового репозитория в удаленный репозиторий
11. Вернитесь в старый клон с репозиторием, выгрузите изменения командой pull с опцией --force
12. Получите все изменения в новом репозитории

Варианты:

|  |  |
| --- | --- |
| № варианта | Репозиторий |
| 1 | https://github.com/liamg/traitor |
| 2 | https://github.com/google/model\_search |
| 3 | https://github.com/gto76/python-cheatsheet |
| 4 | https://github.com/ripienaar/free-for-dev |
| 5 | https://github.com/pystardust/ytfzf |
| 6 | https://github.com/airbnb/javascript |
| 7 | https://github.com/kettanaito/naming-cheatsheet |
| 8 | https://github.com/Sairyss/domain-driven-hexagon |
| 9 | https://github.com/react-native-camera/react-native-camera |
| 10 | https://github.com/spring-projects/spring-petclinic |

**Контрольные вопросы:**

1. Какие существуют типы систем контроля версий? Приведите примеры к каждому типу.
2. К какому типу систем контроля версий относится Git?
3. Что такое репозиторий Git?
4. Что такое GitHub?
5. Что такое коммит?
6. Что такое ветка в репозитории Git?
7. Что такое тег в репозитории Git?
8. Что такое слияние двух веток?
9. Что такое конфликт в Git? Как его решить и почему они бывают?
10. Как отменить слияние веток, если произошел конфликт?
11. Для чего нужен .gitignore?
12. Что делает команда git status?
13. Что делает команда git add?
14. Что делает команда git log?
15. Что делает команда git diff?
16. Что делает команда git show?
17. Что делает команда git stash?
18. Как добавить новую директорию в Git?
19. Что сделает команда "git branch" без какого-либо параметра?
20. Чем отличаются команды "git push" и "git pull"?
21. Что означает статус файла untracked в выводе команды git status?
22. Что означает статус файла new в выводе команды git status?
23. Что означает статус файла modified в выводе команды git status?
24. Чем отличается master и origin/master?
25. Как узнать, кто автор строчки в файле, используя систему Git?
26. Как отменить действие команды "git add" на файл?
27. Как привести измененный файл в начальное состояние (до изменения)?
28. Как сделать ветку с названием my\_branch?ф
29. Как сделать коммит для ветки my\_branch?
30. Как удалить локальную ветку my\_branch?
31. Как исправить ошиоркбку "fatal: The current branch my\_branch has no upstream branch", возникающую при вводе git push?
32. Как удалить все untracked файлы?
33. Что такое форк репозитория?
34. Что такое pull request?

**Требования к отчету:**

По итогу выполнения практической работы необходимо оформить отчёт, включающий:

1. Титульный лист;

2. Оглавление;

3. Формулировки заданий и скриншоты командной строки с требуемыми действиями для задач курсов<https://githowto.com/> и/или<https://learngitbranching.js.org/?locale=ru_RU> (1 и 2 часть ПР)

5. Скриншоты действий, выполненных согласно варианту из части 3 практической работы (не забудьте указать в отчете вариант и задание), а также скриншоты выполнения общего задания (п.7);

6. Ответы на контрольные вопросы (8 любых);

Отчет о практической работы необходимо загрузить в СДО

**Литература для изучения:**

Git Book https://git-scm.com/book/ru/v2

Настройка и использование GitLab <https://otus.ru/nest/post/864/>

Git на сервере – GitLab [https://git-scm.com/book/ru/v2/Git-%D0%BD%D0%B0-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%B5-GitLab](https://git-scm.com/book/ru/v2/Git-на-сервере-GitLab)

Конфликты слияния в Git <https://www.atlassian.com/ru/git/tutorials/using-branches/merge-conflicts>