

ДИСЦИПЛИНА	<b>ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ</b> (полное наименование дисциплины без сокращений)
ИНСТИТУТ	<b>ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ</b>
КАФЕДРА	<b>МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ И СТАНДАРТИЗАЦИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ</b> (полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	<b>МАТЕРИАЛЫ ДЛЯ ПРАКТИЧЕСКИХ/СЕМИНАРСКИХ ЗАНЯТИЙ</b> (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	<b>ЖМАТОВ ДМИТРИЙ ВЛАДИМИРОВИЧ, МЕЛЬНИКОВ ДЕНИС АЛЕКСАНДРОВИЧ, МИРОНОВ АЛЕКСЕЙ ИГОРЕВИЧ, ОВЧИННИКОВ МИХАИЛ АНДРЕЕВИЧ, ПЕТРЕНКО АЛЕКСАНДР АНАТОЛЬЕВИЧ, ПЕТРОВА АННА АЛЕКСАНДРОВНА, СНЕГИРЕВ ДМИТРИЙ АЛЕКСЕЕВИЧ, ТУМАНОВА МАРИНА БОРИСОВНА, ХЛЕБНИКОВА ВАЛЕРИЯ ЛЕОНИДОВНА, ЧЕРНОВ ЕВГЕНИЙ АЛЕКСАНДРОВИЧ</b> (фамилия, имя, отчество)
СЕМЕСТР	<b>4, 2023-2024</b> (указать семестр обучения, учебный год)

**Задание**  
**для выполнения практической работы № 1**  
**«Системы контроля версий»**  
по дисциплине «Технология разработки программных приложений»

**Часть 1. 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. Системы управления репозиториями**

1. Создайте аккаунты на GitHub.
2. Создайте репозиторий согласно варианту.
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. Номер варианта задания студента равен увеличенному на 1 остатку от целочисленного деления номера студента N по списку в группе,

уменьшенного на 1, на число вариантов  $M = 11$ .

$$V = \text{Ocm}((N-1)/M) + 1$$

Пример.

Пусть у студента Петрова В.А. номер по списку в группе равен  $N=16$ . Максимальный номер задания в таблице (число вариантов)  $M=11$ . Тогда номер задания  $V$  студента Петрова В.А. будет равен остатку от деления

$$V = \text{Ocm}((N-1)/M) + 1 = \text{Ocm}((16-1)/11) + 1 = \text{Ocm}(15/11) + 1 = 4 + 1 = 5$$

Если  $N=26$ ,  $M=11$ , то

$$V = \text{Ocm}((N-1)/M) + 1 = \text{Ocm}((26-1)/11) + 1 = \text{Ocm}(25/11) + 1 = 3 + 1 = 4$$

(16-й студент выполняет задание 5 варианта, 26-й студент выполняет задание 4 варианта).

№ варианта	Задание
1	<ol style="list-style-type: none"><li>1) Клонировать непустой удаленный репозиторий на локальную машину</li><li>2) Создать новую ветку и вывести список всех веток</li><li>3) Произвести 3 коммита в новой ветке</li><li>4) Выгрузить изменения в удаленный репозиторий</li><li>5) Откатить ветку обратно на 2 коммита (в том числе в удаленном репозитории)</li><li>6) Вывести в консоли различия между веткой master и новой веткой</li><li>7) Слить новую ветку с master при помощи merge</li></ol>
2	<ol style="list-style-type: none"><li>1) Клонировать непустой удаленный репозиторий на локальную машину</li><li>2) Создать новую ветку и вывести список всех веток</li><li>3) Произвести коммит в ветке master</li><li>4) Произвести 3 коммита в новой ветке в разные файлы</li><li>5) Выгрузить изменения в удаленный репозиторий</li><li>6) Откатить ветку обратно на 2 коммита (в том числе в удаленном репозитории)</li><li>7) Вывести в консоли различия между веткой master и новой веткой</li><li>8) Перебазировать новую ветку на master</li></ol>
3	<ol style="list-style-type: none"><li>1) Клонировать непустой удаленный репозиторий на локальную машину</li><li>2) Создать тег указывающий на последний коммит в ветке</li></ol>

	<p>master</p> <p>3) Создайте новую ветку и выведите список всех веток</p> <p>4) Произведите 3 коммита в новой ветке</p> <p>5) Выгрузите все изменения в удаленный репозиторий</p> <p>6) Откатите ветку к созданному тегу (в том числе в удаленном репозитории)</p> <p>7) Выведите в консоли различия между веткой master и новой веткой</p>
4	<p>1) Клонировать непустой удаленный репозиторий на локальную машину</p> <p>2) Создайте новую ветку и выведите список всех веток</p> <p>3) Произведите 3 коммита в новой ветке в разные файлы</p> <p>4) Выгрузите изменения в удаленный репозиторий</p> <p>5) Откатите в новой ветке предпоследний коммит (в том числе в удаленном репозитории)</p> <p>6) Выведите в консоли различия между веткой master и новой веткой</p> <p>7) Слейте новую ветку с master при помощи merge</p>
5	<p>1) Клонировать непустой удаленный репозиторий на локальную машину</p> <p>2) Создайте новую ветку и выведите список всех веток</p> <p>3) Произведите 3 коммита в новой ветке в разные файлы</p> <p>4) Выгрузите изменения в удаленный репозиторий</p> <p>5) Произведите revert предпоследнего коммита в новой ветке</p> <p>6) Выведите в консоли различия между веткой master и новой веткой</p> <p>7) Слейте новую ветку с master при помощи merge</p>
6	<p>1) Клонировать непустой удаленный репозиторий на локальную машину</p> <p>2) Создайте новую ветку и выведите список всех веток</p> <p>3) Произведите 3 коммита в новой ветке в новых файлах</p> <p>4) Выгрузите изменения в удаленный репозиторий</p> <p>5) Произведите какие-нибудь изменения в файле который существует в ветке master, но не коммитьте их</p> <p>6) Выполните git stash</p> <p>7) Переключитесь на ветку master</p> <p>8) Выполните git stash pop</p> <p>9) Выведите в консоли различия между веткой master и новой веткой</p>

	10) Слейте новую ветку с master при помощи merge
7	1) Клонировать 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) Клонировать непустой удаленный репозиторий на локальную машину 2) Создайте две новых ветки и выведите список всех веток 3) Произведите по 2 коммита в обеих 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) Клонировать 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) Клонировать 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) Клонировать 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

### **Часть 3. Работа с ветвлением и оформление кода**

Продолжите проходить курс на сайте <https://githowto.com/> (начиная с раздела 22).

Создайте и разрешите конфликты в репозитории, согласно варианту:

1. Сделайте форк репозитория в соответствии с Вашим вариантом.

Номер варианта задания студента равен увеличенному на 1 остатку от целочисленного деления номера студента N по списку в группе, уменьшенного на 1, на число вариантов M = 10.

$$V = \text{Ост}((N-1)/M) + 1$$

Пример.

Пусть у студента Петрова В.А. номер по списку в группе равен N=9. Максимальный номер задания в таблице (число вариантов) M=10. Тогда номер задания V студента Петрова В.А. будет равен остатку от деления

$$V = \text{Ост}((N-1)/M) + 1 = \text{Ост}((9-1)/10) + 1 = \text{Ост}(8/10) + 1 = 8 + 1 = 9$$

Если N=26, M=10, то

$$V = \text{Ост}((N-1)/M) + 1 = \text{Ост}((26-1)/10) + 1 = \text{Ост}(25/10) + 1 = 5 + 1 = 6$$

(9-й студент выполняет задание 9 варианта, 26-й студент выполняет задание 6 варианта).

2. Склонировать его (форк репозитория) на локальную машину.
3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e.
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 при этом конфликты.
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 с опцией --force
12. Получите все изменения в новом репозитории

### Варианты

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

### 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 коммит?
5. Что такое ветка в репозитории Git?
6. Что такое тег в репозитории Git?
7. Что такое слияние двух веток?
8. Что такое конфликт в Git? Как его решить и почему они бывают?
9. Как отменить слияние веток, если произошел конфликт?
10. Для чего нужен .gitignore?

11. Что делает команда `git status`?
12. Что делает команда `git add`?
13. Что делает команда `git log`?
14. Что делает команда `git diff`?
15. Что делает команда `git show`?
16. Что делает команда `git stash`?
17. Как добавить новую директорию в Git?
18. Что сделает команда `"git branch"` без какого-либо параметра?
19. Чем отличаются команды `"git push"` и `"git pull"`?
20. Что означает статус файла `untracked` в выводе команды `git status`?
21. Что означает статус файла `new` в выводе команды `git status`?
22. Что означает статус файла `modified` в выводе команды `git status`?
23. Чем отличается `master` и `origin/master`?
24. Как узнать, кто автор строчки в файле, используя систему Git?
25. Как отменить действие команды `"git add"` на файл?
26. Как привести измененный файл в начальное состояние (до изменения)?
27. Как сделать ветку с названием `my_branch`?
28. Как сделать коммит для ветки `my_branch`?
29. Как удалить локальную ветку `my_branch`?
30. Как исправить ошибку `"fatal: The current branch my_branch has no upstream branch"`, возникающую при вводе `git push`?
31. Как удалить все `untracked` файлы?
32. Что такое GitHub?
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 курса <https://githowto.com/> (1 и 2 часть ПР) или обязательные 11 пунктов из задания;
4. Скриншоты действий, выполненных согласно варианту из частей 2 и 3 практической работы (не забудьте указать в отчете вариант и задание), а также скриншоты выполнения общего задания;
5. Ответы на контрольные вопросы (8 любых) с развернутым пояснением;
6. Вывод о проделанной работе.

Отчет о практической работе необходимо загрузить на сайт ЦДО.

### **Литература для изучения**

[Git Book](#)

[Конфликты слияния в Git](#)

[Learn Git Branching](#)