**Вопросы**

**1. Что такое система контроля версий, для чего ее используют?**

*программное обеспечение для облегчения работы с изменяющейся информацией и разработки проекта совместно с коллегами.*

⎫ автоматическое создание архива (бэкап) для синхронизации кодовой базы; ⎫ отслеживание изменений (кто, когда и зачем сделал изменения); ⎫ совместная работа над одним и тем же проектом; ⎫ отслеживание ошибок (Bug трекинговая система).

**2. Какой принцип хранения файлов использует Git?**

*Git хранит информацию в структуре данных, называемой* ***репозиторий****.*

Репозиторий, хранилище – место, где хранятся и поддерживаются какие-либо данные.

*// Git хранит данные сходным с файловыми системами UNIX способом, но в немного упрощённом виде.*

**3. В чем отличие Git от других систем контроля версий?**

*Хранит снимки состояний, а не изменений. Локальность операций. Целостность.*

**4. В каких трех основных состояниях файлы могут находиться в Git ?**

***Модифицированное (modified)*** *– изменения уже внесены в файл, но пока не зафиксированы в базе данных.* ***Индексированное (staged)*** *– текущая версия модифицированного файла помечена как требующая последующей фиксации.* ***Зафиксированное (committed)*** *– данные надежно сохранены в локальной базе.*

**5. Что такое индексация файла? Какой командой она выполняется?**

*Это процесс добавления текущего содержимого (изменённого) файла в индекс.*

*Команда - $ git add \*имя файла\**

**6. Что такое фиксация файла? Какой командой она выполняется?**

*Окончательное добавление в репозиторий, когда git запоминает файл навсегда и следит за всеми последующими изменениями.*

*Команда - $ git commit -m "added \*имя файла\* to the repo".*

**7. Продемонстрируйте команду проверки выбранных настроек.**

Git config - list

**8. Как инициализировать репозиторий в существующей папке?**

Git init

**9. Как указать файлы, за которыми должна следить система?**

Git add имяфайла

**10.Как выполнить фиксацию изменений?**

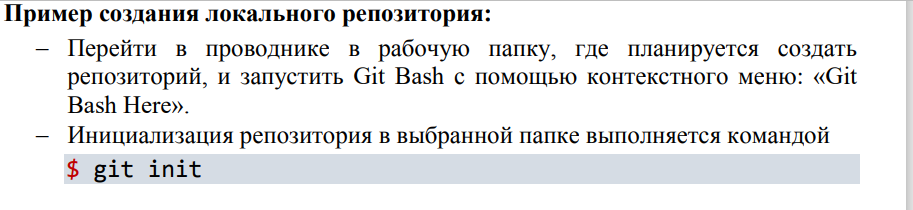
Git commit -m “added имяфайла to the repo”

**11.Какой командой определить состояния файлов?**

Git status

**12.Для чего создается файл .gitignore? Поясните его структуру**

*для того, чтобы определить, какие файлы и папки не нужно добавлять в git репозиторий. Это простой текстовый файл, в каждой строке которого содержится шаблон, который файлы или каталоги следует игнорировать*



git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы

git commit – выполняет коммит проиндексированных файлов в репозиторий

изменения можно зафиксировать в репозитории командой:

$ git commit -m "added Hello.txt to the repo" Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo")

*.*

**Вопросы.**

1. Что такое система контроля версий, для чего ее используют?

2. Какой принцип хранения файлов использует Git?

3. В чем отличие Git от других систем контроля версий?

4. В каких трех основных состояниях файлы могут находиться в Git ?

5. Что такое индексация файла? Какой командой она выполняется?

6. Что такое фиксация файла? Какой командой она выполняется?

7. Продемонстрируйте команду проверки выбранных настроек.

8. Как инициализировать репозиторий в существующей папке?

9. Как указать файлы, за которыми должна следить система?

10.Как выполнить фиксацию изменений?

11.Какой командой определить состояния файлов?

12.Для чего создается файл .gitignore? Поясните его структуру

**Вопросы.**

1. **Как используется команда git log?**

Просмотреть журнал можно с помощью команды $ git log которая показывает лог commits начиная с указателя HEAD

Полная история репозитория(все коммиты,и т.д.);

1. **Что такое ветвление?**

Ветвление (branching) означает отклонение от основной линии разработки, после которого работа перестает затрагивать основную линию и переходит в ветвь.

1. **Как используется команда git checkout?**

Переключение веток осуществляется командой: **git checkout**

1. **Что такое указатель HEAD?**

Указатель на текущую ветку и на последний коммит;

1. **Продемонстрируйте создание новых веток в Git?**

git branch <name>;

1. **Какая команда осуществляет переключение веток?**

git checkout <name>;

1. **Для чего используется команда merge?**

**Команда** **git** **merge** выполняет **слияние** отдельных направлений разработки, созданных с помощью **команды** **git** branch, в единую ветку.

1. **Какая команда позволяет отобразить историю репозитория?**

$ git log

1. **Как выполнить слияние изменений из одной ветки в другую?**

**$ git merge <сливаемая ветка>**

**10.Какие проблемы могут быть при слиянии и как они разрешаются**

*различия текста в файлах веток (несовпадающие места в файле для двух веток). Конфликт разрешается разработчиком вручную.*