**GITHUB**

СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT  
  
1.Опишите назначение систем CVS.  
2. Какие виды CVS вам известны? Какими особенностями они обладают?  
3. На примере системы контроля версиями GIT опишите особенности подхода к работе с данными. Какими преимуществами данная система обладает?  
4. Опишите процесс установки И настройки GIT на свой ПК .

1. Опишите назначение сервиса GitHub.  
2. Опишите назначение сервиса Github Desktop.  
3. Поясните термины git и github:  
1. Репозиторий  
2. Рабочая область и хранилище  
3. Форк  
4. Ветка  
5. Мастер  
6. Коммит  
7. Пул  
8. Пуш  
9. Пулрекквест  
10. Мердж  
11. Кодревью

СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT  
1. CVS (Concurrent Versions System) - это система контроля версий файлов. Основное предназначение - сохранение истории изменения файлов. (В качестве примеров в дальнейшем будут использоваться исходные тексты на языке С.) Необходимость в системе контроля версий появляется например тогда, когда в достаточно большом проекте ошибка проявляется только через некоторое время после внесения изменений.  
2. Системы контроля версий бывают локальными, централизованными или распределёнными.  
• Локальная система хранит файлы на одном устройстве, централизованная использует общий сервер, а распределённая — общее облачное хранилище и локальные устройства участников команды. В локальной системе удобно работать с большими проектами, но сложно взаимодействовать с удалённой командой.  
• В централизованной системе налажена удалённая работа, но всё привязано к одному серверу. Любой сбой или взлом может повредить файлы проекта.  
• В распределённой системе налажена удалённая работа. Если с файлами основного репозитория что-то случится — проект легко восстановить из копии любого участника команды.  
3. Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными. Подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён. Git представляет свои данные как, скажем, поток снимков.  
4. 1) Переходим на официальный сайт Git  
2) Выбираем нужную версию для вашей ОС.  
3) Запускаем установочный файл и устанавливаем Git.  
4) Открываем терминал (Linux и MacOS) или консоль (Windows) и вводим следующие команды:  
git config —global [user.name](http://user.name/) "<ваше\_имя>" - установка имени  
git config —global [user.email](http://user.email/) "<[адрес\_почты@email.com](http://%D0%B0%D0%B4%D1%80%D0%B5%D1%81_%D0%BF%D0%BE%D1%87%D1%82%D1%8B@email.com/)>" - установка e-mail  
GITHUB  
1. GitHub — это хостинг: он позволяет хранить проекты удалённо на сервере и работать с ними из любой точки мира. Доступ к файлам есть у всех, у кого есть ссылка. Одна из главных функций GitHub — контроль версий. Все изменения в коде можно отследить, поэтому в командной разработке это незаменимая вещь.  
2. GitHub Desktop – программа для удобной работы с проектами, размещенными на одноименном сервисе. С ее помощью можно добавить любые репозитории сразу на жесткий диск, обычным перетаскиванием или через проводник. Если перетаскиваемая папка является Git репозиторием, то GitHub импортирует всю историю и подсоединит её к профилю пользователя.  
3. Git (/ɡɪt/, Global Information Tracker) - это программное обеспечение для отслеживания изменений в любом наборе файлов, обычно используемое для координации работы программистов, совместно разрабатывающих исходный код во время разработки программного обеспечения.  
GitHub — это служба размещения в Интернете репозиториев Git.  
4. Репозито́рий (от англ. repository — хранилище) — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.  
5. Корневая папка проекта — это рабочая область. В ней находятся все файлы и папки, необходимые для его работы. Хранилище — это содержимое скрытой папки .git. В этой папке хранятся все версии рабочей области и служебная информация.  
6.Форк — полная копия мастер-репозитория, в которой вы можете безопасно работать.  
7. В Git ветки — это элемент повседневного процесса разработки. По сути ветки в Git представляют собой указатель на снимок изменений. Если нужно добавить новую возможность или исправить ошибку (незначительную или серьезную), вы создаете новую ветку, в которой будут размещаться эти изменения.  
8. Мастер - ветка по умолчанию. Когда идёт создание коммитов на начальном этапе дана ветка master, указывающая на последний сделанный коммит. При каждом новом коммите она сдвигается вперёд автоматически.  
9. Коммит — это снимок локальных файлов, записанный в локальный репозиторий.  
10. Git pull используется для извлечения и загрузки содержимого из удаленного репозитория и немедленного обновления локального репозитория этим содержимым.  
11. Git push используется для выгрузки содержимого локального репозитория в удаленный репозиторий.  
12. Пул-реквест — это запрос (англ. request — «запрос») на интеграцию изменений из одной ветки в другую.  
13. Git merge выполняет слияние отдельных направлений разработки, созданных с помощью команды git branch, в единую ветку.  
14.Code Review — это процесс проверки и анализа кода задачи разработчиком перед ее релизом.