**Контрольные вопросы по СКВ**

**Вариант 1**

**Николаев Денис 3ИП-1-22**

*Ответить на вопросы, редактируя данный документ*

*Создать репозиторий в папке с данным документом*

*Опубликовать его в удаленном репозитории (ваш собственный public репозиторий на github.com)*

*Ссылку на этот репозиторий прикрепить к заданию в качестве ответа*

1. **Что такое система контроля версий и зачем она нужна?**

Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии. Для контроля версий файлов в качестве примера можно привести исходный код программного обеспечения, хотя можно использовать контроль версий практически для любых типов файлов.

1. **Объясните различие между локальным репозиторием и удаленным репозиторием в Git.**

Локальный расположен непосредственно в памяти компьютера разработчика, в нем происходит разработка и фиксация изменений, после чего можно отправить на удаленный репозиторий, а удалённый находится на сервере, может быть приватным – доступным ограниченному числу лиц, и публичным – open source.

1. **Что такое commit в Git и какова его роль?**

Commit’ы делает для проекта снимок текущего состояния изменений, добавленных в раздел проиндексированных файлов. Такие подтвержденные снимки состояния можно рассматривать как «безопасные» версии проекта — Git не будет их менять, пока вы явным образом не попросите об этом.

1. **Как создать новый репозиторий в Git и что происходит при инициализации?**

Создать новый репозиторий можно через серию команд (в пример взят репозиторий KontrVoprSKV):

echo "# KontrVoprSKV" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/D3n4ik132/KontrVoprSKV.git

git push -u origin main

Инициализация позволяет создать структуру, в которой будут храниться все изменения и история проекта. После инициализации репозитория вы можете начать добавлять файлы, фиксировать изменения и управлять ветками.

1. **Что такое ветвление в Git и как создать новую ветку?**

Почти каждая система контроля версий в той или иной форме поддерживает ветвление. Используя ветвление, Вы отклоняетесь от основной линии разработки и продолжаете работу независимо от неё, не вмешиваясь в основную линию. Во многих системах контроля версий создание веток — это очень затратный процесс, часто требующий создания новой копии каталога с исходным кодом, что может занять много времени для большого проекта.

Новые ветки создаются с помощью команды git branch *имя\_ветки*