

Задача за самостоятелна работа.

Георги, Петър и Иван работят по даден проект, като използват GitHub система за контрол на версиите.

Въпрос 1: Какъв модел за контрол на версиите използват?

GitHub използва разпределен модел за контрол на версиите (Distributed Version Control System, DVCS). В разпределената система за контрол на версиите, всеки участник клонира цялото хранилище, включително цялата история на проекта, и работи локално.

Георги създава собствено Repository и трябва да присъедини към него Иван и Петър.

Въпрос 2: Как може да се осъществи това присъединяване? (опишете стъпките по присъединяване)

За да присъедини Георги, Иван и Петър към своето хранилище, той трябва да следва следните стъпки:

- Отвори хранилището (repository) в GitHub.

- Натисне на таба "Settings" в хранилището.

- Отиде на секцията "Manage access -> Invite a collaborator".

- Введе потребителските имена или имейлите на Иван и Петър в полето за търсене.

- Избере Иван и Петър от списъка и натисне "Add collaborator".

- Иван и Петър е необходимо да си проверят e-mail-а, да потвърдят (Accept Invitation)

Всички решават да работят по възложеният им проект, като използват среда за разработка IntelliJ IDEA.

Въпрос 3: Опишете стъпката/стъпките, които те трябва да направят за да използват Repository-то, създадено по-рано?

За да използват хранилището, създадено от Георги, в IntelliJ IDEA, те трябва да направят следното:

- Отворят IntelliJ IDEA и създадат нов проект или отворят съществуващ проект. Натискат на "VCS" в горното меню и избират "Checkout from Version Control", а след това "Git".

- Въвеждат URL на хранилището, което Георги е създал, в полето за URL и натискат "Clone".

- Ако се налага, въвеждат своите данни за аутентикация в GitHub.

- След клонирането, проектът ще бъде достъпен в IntelliJ IDEA за работа.

Въпрос 4: Колко инстанции на хранилището ще имат те по своя проект?

Те ще имат общо една инстанция на хранилището отдалечено на GitHub и по една локална инстанция на хранилището на всяка от техните машини. Това са общо четири инстанции – една на сървъра (GitHub) и по една локална за Георги, Иван и Петър.

На локалната машина Георги започва своята работа по проекта. Георги създава файловете А и В (текстови файлове). След известно време той решава, че е нужно да запази промените, които е направил така, че Иван и Петър да може да видят и тези промени.

Въпрос 5: Чрез каква/какви стъпка/стъпки Георги запазва промените, които е направил по файловете, по които е работил? Запишете и съответната/съответните изпълнявана/изпълнявани команда/команди.

`git add A B` - Тази команда добавя файловете А и В към списъка с промени за следващия commit.

`git commit -m "Add files A and B"` - Създава commit със съобщението, описващо промените.

`git push` - Изпраща commit-а от локалното хранилище към отдалеченото хранилище в GitHub, където Иван и Петър могат да го видят.

В допълнение преди push командата може да се направи нов branch с командата `git checkout -b new-files-a-b` и след това да се push-не към отдалеченото хранилище

Малко по-късно Иван също решава да работи по проекта. Той вижда направените файлове от Георги и започва работа по един от тях. След като приключва с работата си, Иван решава да запази своите промени. В последствие Иван разбира, че няма да може да запази промените си отдалечено, а само локално.

Въпрос 6: Каква може да е причината Иван да не може да запази работата си отдалечено? (Дайте поне една причина).

Има конфликт с промени, които вече са били публикувани от друг участник, и Иван трябва първо да синхронизира локалното си хранилище с отдалеченото.

Въпрос 7: Чрез каква команда Иван запазва локално промените си?

За да запази Иван локално промените си, той може да използва следната команда: `git commit -am "Съобщение за промените"`. Тази команда добавя към следващия commit всички промени по файлове, които вече са били добавени в хранилището (т.е. файлове, които са следени от Git) и създава commit с предоставеното съобщение. Ако Иван работи върху нови файлове, които преди това не са били добавени към хранилището, той трябва първо да изпълни `git add <име на файл>` за всеки нов файл, преди да направи commit. Commit-ът създава точка в историята на проекта, към която можем да се върнем по всяко време. Това е начинът, по който запазваме нашата работа в Git.

Въпрос 8: Какво трябва да направи Петър, за да види промените направени от Георги? Ще види ли Петър промените, направени от Иван?

За да види Петър промените направени от Георги, той трябва да използва командата `git pull` за да актуализира своята локална копия с последните промени от отдалеченото хранилище. Командата `git pull` комбинира `git fetch`, която изтегля промените от отдалеченото хранилище, и `git merge`, която слива тези промени в текущата работна директория. Петър няма да види промените направени от Иван, тъй като Иван е запазил своите промени само локално и не е успял да ги публикува в отдалеченото хранилище. За да види Петър промените на Иван, Иван първо трябва да разреши всякакви проблеми, които му пречат да публикува своите промени, и след това да изпълни `git push` за да ги качи в отдалеченото хранилище. След това Петър може да използва `git pull` за да види и тези промени.