

Lucru de laborator nr. 1

Tema lucrării laboratoare:

Pregătirea inițială a mediului Git. Crearea unui repository pe Git și unul local.

Partea teoretică:

Git este un sistem de control distribuit al versiunilor, gratuit și cu sursă deschisă, conceput pentru a lucra rapid și eficient cu orice proiect, de la cel mic la cel foarte mare. Git este ușor de învățat, ocupă puțin spațiu și are o performanță fulminantă. Depășește instrumentele SCM (Supply Chain Management), cum ar fi Subversion, CVS, Perforce și ClearCase, cu caracteristici precum ramificarea locală cu costuri reduse, zone de stocare convenabile și fluxuri de lucru multiple.

Ramificare și fuziune

Caracteristica lui Git care îl diferențiază cu adevărat de aproape toate celelalte SCM-uri este modelul de ramificare. Git permite și încurajează crearea mai multor ramuri locale care pot fi complet independente una de cealaltă. Crearea, fuzionarea și ștergerea acestor ramuri durează câteva secunde. Acest lucru înseamnă că puteți face lucruri precum:

- Schimbarea contextului fără contact. Creați o ramură pentru a încerca o idee, faceți câteva confirmări, întoarceți-vă la locul de unde a fost creată ramura, aplicați corecția, întoarceți-vă la locul unde ați experimentat și fuzionați-o.

- Codeline de rol. Aveți o ramură care conține întotdeauna doar ceea ce este trimis în producție, o alta care fuzionează munca pentru testare și mai multe ramuri mai mici pentru munca de zi cu zi.

- Flux de lucru bazat pe caracteristici. Creați ramuri noi pentru fiecare nouă caracteristică la care lucrați, astfel încât să puteți trece fără probleme de la una la alta, iar apoi ștergeți fiecare ramură atunci când caracteristica este fuzionată în linia principală.

- Experimente unice. Creați o ramură pentru experimente, vă dați seama că nu funcționează și pur și simplu o ștergeți, renunțând la munca dvs. și fără ca nimeni altcineva să o vadă (chiar dacă ați promovat alte ramuri între timp).

Distribuit

Una dintre cele mai bune caracteristici ale oricărui SCM distribuit, inclusiv Git, este că este distribuit. Acest lucru înseamnă că, în loc să efectuați o "editare" a verigii curente de cod sursă, efectuați o "clonare" a întregului depozit.

Copii de rezervă multiple

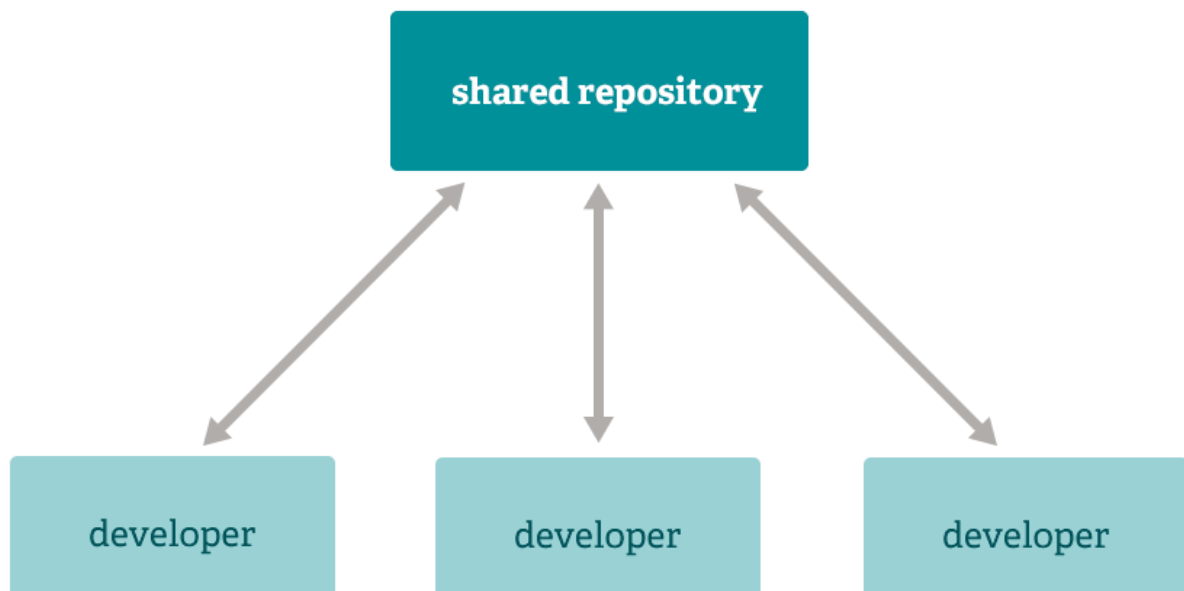
Acest lucru înseamnă că, chiar dacă utilizați un flux de lucru centralizat, fiecare utilizator are, în esență, o copie de rezervă completă a serverului principal. Fiecare dintre aceste copii poate fi adusă pentru a înlocui serverul principal în caz de defecțiune sau de corupție. În esență, nu există un singur punct de eșec în Git decât dacă există o singură copie a depozitului.

Orice flux de lucru

Datorită naturii distribuite a lui Git și a sistemului superior de ramificare, este posibil să se implementeze un număr aproape infinit de fluxuri de lucru cu o relativă ușurință.

Flux de lucru de tip Subversion

Fluxurile de lucru centralizate sunt foarte frecvente, în special pentru persoanele care se mută de la un sistem centralizat. Git nu vă va permite să faceți push dacă cineva a făcut un push de la ultima verificare, așa că un model centralizat în care toți dezvoltatorii fac push pe același server funcționează foarte bine.



Sursa : <https://git-scm.com/>

Instalarea Git

Descărcați programul de instalare (<https://git-scm.com/downloads>), rulați programul de instalare, selectați opțiunile pe care le doriți (sau lăsați-l așa cum este dacă nu știți prea multe despre el), așteptați ca procesul de instalare să se încheie și ați terminat.

Utilizarea Git

În folderul de lucru în care se află fișierele sursă ale aplicației/proiectului dumneavoastră, prin intermediul terminalului sau al liniei de comandă, utilizați comanda git init (documentația oficială - <https://git-scm.com/docs/git-init>) pentru a inițializa depozitul local, după care Git va urmări modificările aduse fișierelor sursă și pentru a confirma aceste modificări - utilizați comanda git commit (modificări - utilizați comanda git commit (<https://git-scm.com/docs/git-commit>). În viitor, dacă va fi necesar să se sincronizeze modificările din depozitul partajat - se utilizează comanda git push

(<https://git-scm.com/docs/git-push>).

Pentru a actualiza fișierele proiectului, se utilizează git merge cu comanda git pull. Deoarece, în cazul în care un alt membru actualizează depozitul, utilizarea comenzii git push va genera o eroare.

Sarcină principală:

1. Instalați mediul Git pe dispozitivul dumneavoastră, pregătiți un dosar pentru aplicația/proiectul dumneavoastră și inițializați depozitul local în acest dosar (comanda git init).
2. Implementați o aplicație simplă. În cadrul unei echipe, stabiliți cine va implementa ce parte a aplicației, care va fi acea parte și ce funcție (sau funcții) va îndeplini.
3. Efectuați comenzi de confirmare a modificărilor în codul aplicației/proiectului (comanda git commit). Creați un cont GitHub (dacă nu aveți unul) și trimiteți numele de utilizator sau link-ul profilului către instructor.
4. După ce primiți instrucțiuni suplimentare de la instructor, împingeți modificările în ramura dvs. separată din depozitul GitHub (sau, dacă ați fost de acord să vă implementați propriul proiect, într-un depozit separat pentru proiect) (comanda git push).

Variante:

- O aplicație care efectuează operații aritmetice simple, cum ar fi adunarea, scăderea, înmulțirea, împărțirea, adunarea la grad și extragerea rădăcinilor. Fiecare membru al

echipei implementează o clasă care efectuează una sau două operații aritmetice. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.

- O aplicație care extrage informații specifice din textul introdus. Fiecare membru al echipei implementează o clasă care extrage în mod specific informațiile necesare. (notă: încercați să folosiți expresii regulate) În algoritmul principal (clasa), realizați relația dintre cel puțin două clase. (notă: încercați să folosiți expresii regulate).
- O aplicație care manipulează textul - îndepărtează spațiile, introduce o cratimă pe o linie nouă, oglindește textul, separă cuvinte, schimbă cuvinte, schimbă litere în cuvinte. Fiecare membru al echipei implementează o clasă care efectuează manipulări specifice de text. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație care realizează manipularea fișierelor - exemplele de manipulare sunt similare din varianta anterioară. Fiecare membru al echipei implementează o clasă care efectuează manipulări specifice de fișiere. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație care calculează proprietăți geometrice, cum ar fi aria și perimetrul și raza pentru diferite forme (de exemplu, pătrate, dreptunghiuri, cercuri, triunghiuri). Fiecare membru al echipei implementează o clasă care calculează o proprietate specifică pentru o anumită formă. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație care recomandă rețete pe baza preferințelor utilizatorului și a ingredientelor disponibile. Fiecare membru al echipei implementează o clasă care să recomande rețete dintr-o anumită bucătărie sau categorie. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație pentru conversia diferitelor monede. Fiecare membru al echipei implementează o clasă care se ocupă de conversia într-o anumită monedă. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație care simulează aruncarea zarurilor și urmărește rezultatele. Fiecare membru al echipei poate implementa o clasă care să simuleze un anumit tip de zaruri (de exemplu: cu șase fețe, cu douăsprezece fețe). În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.
- O aplicație care simulează selectarea fantomelor (fiecare fantomă este o clasă). Pe baza parametrilor introduși de utilizator (parametrii sunt numere sau text), aplicația va produce un anumit tip de fantomă. Fiecare membru al echipei trebuie să implementeze

o clasă pentru fantomă cu trei parametri de definit. În algoritmul principal (clasa), realizați relația dintre cel puțin două clase.

După finalizarea lucrării, redactați un raport, care trebuie să includă - numele, prenumele, grupul, tema lucrării practice, varianta dvs. de implementare a sarcinii, o scurtă descriere a implementării sarcinii, un link către codul sursă pe GitHub. Salvați raportul în format PDF și trimiteți-l la ELSE.