



Versionshantering med Git

Inledande Programmering med Java



Planering för eftermiddagen

01 Introduktion

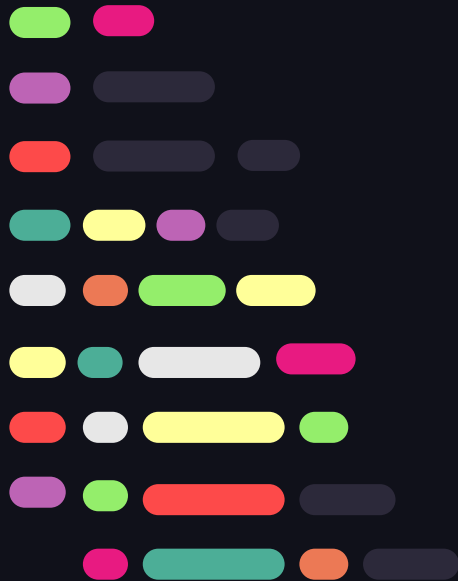
Introduktion till versionshantering

02 Konfiguration

Vi kikar på vad som behövs för att
Git ska fungera

03 Övningar

Vi kommer testa att kлона repositories och
sedan göra lite gemensamma övningar i
debugging och kodförståelse





Vad är versionshantering?

- Versionshantering ser till att kod inte försvinner om din dator kraschar
- Låter flera utvecklare arbeta med samma kodbas samtidigt
- Du kan återställa projektet till en tidigare version om allting skiter sig
- Är ett verktyg som underlättar, inte försvårar

Om Git

- Finns mer än ett system för versionshantering, men Git är det mest populära
- Skapades av Linus Thorvalds som också skapade Linux
- Har integrerad support i moderna IDE som IntelliJ och Visual Studio



{

“It is easy to shoot your foot off with git, but also easy to revert to a previous foot and merge it with your current leg”

- Jack William Bell

}



Förebygga plagiat

- Inte bara målet (er lösning på uppgiften) som räknas utan vägen dit är minst lika viktig (hur ni gjort för att lösa uppgiften)
- AI-verktyg gör det lätt att skapa kod men svårare att spåra processen och förstå bakgrunden, vilket är en anledning att ta som vana att göra små ändringar i varje version och kommentera dessa med en kort beskrivning
- Lär från misstag och undvik att göra om samma sak igen men möjligheten finns också att gå tillbaka till en tidigare version

Med versionshantering via Git får koden författare som är ansvariga för ändringarna och skapar på så vis spårbarhet, alltså är det möjligt att se vem som gjort vad om ni samarbetar



Konfigurera Git

Kontrollera att Git är installerat genom
att öppna en command prompt och skriva:

```
git --version
```





Konfigurera Git

**Konfigurera användarnamn och email
(ersätt med egna namn och adresser tack! :))**

```
git config --global user.name "Carl-Johan"
```

```
git config --global user.email "carljohan@im.uu.se"
```

Använd helst en mail som ni kommer åt även efter utbildningen



{

Att använda Git med IntelliJ

}



INTERMISSION

Kaffepaus



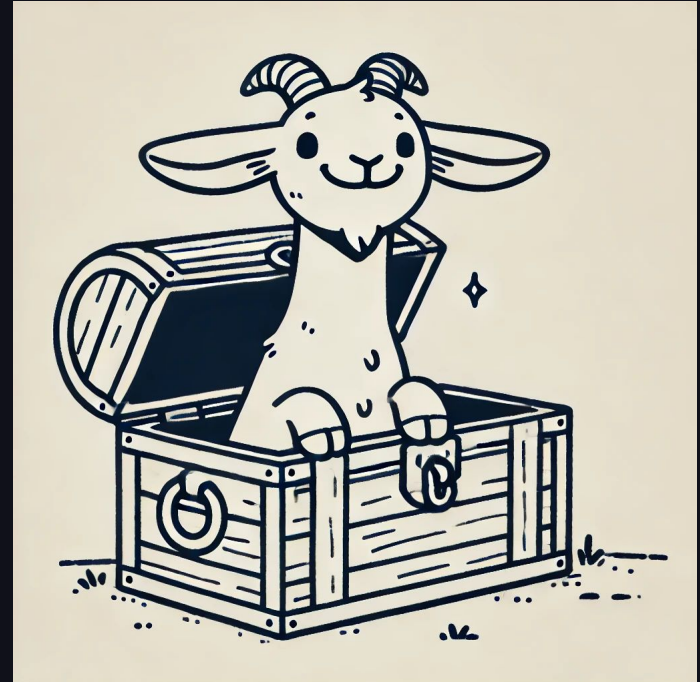
Klona repository

- Starta IntelliJ
- Stäng eventuella öppna projekt
- Välj knappen “Get from VCS”/”Clone repository” intill “New project” och “Open”
- Skriv in följande länk i URL-fältet:
<<https://github.com/lovisajoh/monty-hall-debugging-exercise>>
- Välj var projektet ni hämtar ska sparas
- Tryck på Clone

Monty Hall-problemet



- Tre dörrar: bakom en av dem finns ett pris. Bakom de andra finns en get
- Spelaren väljer en dörr. Monty öppnar sedan en av de andra dörrarna som innehåller en get och frågar om spelaren vill byta dörr
- Om man byter har man större chans att vinna





Tips på arbetsflöde

- Börja med att se till att koden går att läsa
- > Se sedan till att den går att kompilera
- > Fixa därefter körtidsfelen så att programmet inte kraschar
- > Fixa sen de logiska felen så att programmet gör rätt saker
- > Fundera på namnsättning och kommentarer



Checklista

- Klassnamn samma som filnamn, annars refactoring
- Namnsättning enligt Javas namnstandard (engelska namn: verb för metoder, substantiv obestämd form för klasser)
- Logisk kod
- Indentering (använd refactoring)
- Beskrivande kommentarer för vad som görs (inte hur)
- JavaDoc-format
- Lyssna på varningar och intellisense
- Använd rätt datatyp för rätt sak
- Optimera import
- Åtkomst



Bra länkar

- [Guide till Github markdown language](#)
- [Bok/manual för Git](#)
- [IntelliJ:s dokumentation för Gitintegrering](#)
- [Youtubelänk: Basic Git and Github with IntelliJ](#) (6 min)





Allmänna tips

- Skriv alltid en egen **.gitignore** för ert projekt i stället för att förlita er på templates (Det här är särskilt viktigt när ni arbetar i grupp)
- Generera aldrig filer när ni skapar ett repository på Github så undviker ni problem där ni råkar pusha filer till olika branches
- Om ni behöver göra en rollback någon gång så läs noga om vad de olika alternativen innebär först
- Lär er att använda git via terminalen också när ni börjar bli bekväma med IntelliJ-integrationen





Allmänna tips

- Dela inte lösningarna till uppgifterna med andra
- Ha som vana att göra Git-projekt (Repositories) privata när ni arbetar med uppgifter på javakursen.

GitHub är en öppen plattform där utvecklare kan ta del av andras material för att kunna lära av varandra vilket ni får göra fritt med era egna idéer.

