NTNU

Kunnskap for en bedre verden

Kollokvium 3 - Oppgaver

TDT4100 Objektorientert programmering

Mandag 24.02.2020

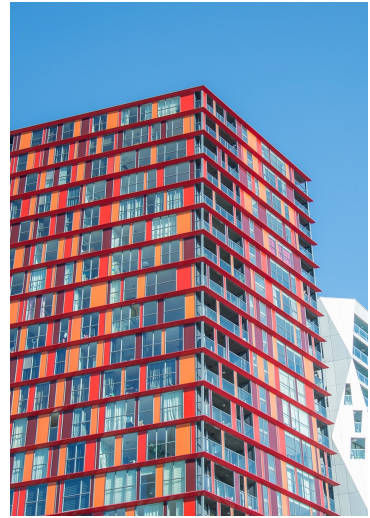
NTNU

Oppgaver

I dette kollokviet skal vi skrive videre på koden fra øvingsforlesningen som skulle representere en bedrift. Bedriften skal representeres som en gruppe avdelinger, som seg igjen kan ha underavdelinger, osv. Hver avdeling kan ha et sett med ansatte tilknyttet seg.

I tillegg skal vi se på testing. Vi skal skrive noen JUnit tester for å teste koden vi skriver i kollokviet.

Utdelt er filene **Department.java** og **DepartmentTest.java** som er koden som ble skrevet i øvingsforlesningen.



NTNU

Pakker

Utdelt kode og oppgavene finnes i mappen foreksempel/src/kollokvie3.

foreksempel/src/kollokvie3.kode

Her finner dere utdelt kode og kan skrive deres egen kode.

foreksempel/src/kollokvie3.underveis

Her blir kode lagt inn rett etter felles gjennomgang.

foreksempel/src/kollokvie3.tester

Her ligger kode som kan brukes til å teste at en oppgave er løst riktig

foreksempel/src/kollokvie3.lf

Blir gjort tilgjengelig i etterkant og innholder et løsningsforslag for oppgavene.

NTNU

Oppgave 1

7 min

Motivasjon

Det kan være nyttig å vite om en avdeling er under en annen.

Oppgavetekst

Skriv en metode **contains** i **Department** som tar inn en avdeling og sjekker om den er en underavdeling av avdelingen som **contains** ble kalt på. Her må man både tenke på direkte underavdelinger og underavdelinger av underavdelinger.

Testing

Se neste oppgave.

NTNU

Oppgave 2

7 min

Motivasjon

Det er lett å gjøre noe feil når man driver med rekursjon. Vi skal derfor teste funksjonaliteten vi nettopp skrev.

Oppgavetekst

Skriv en JUnit test som lager avdelinger og sjekker at **contains** fungerer ordentlig, både for direkte underavdelinger, indirekte underavdelinger og avdelinger som ikke er underavdelinger av hverandre.
Tips: ***assertTrue*** *og* ***assertFalse*** *funksjonen kan være nyttige*

Testing

Kjør JUnit testen du nettopp har skrevet, og sjekk at den ikke feiler.

NTNU

Oppgave 3

7 min

Motivasjon

Hvor en avdeling tilhører kan endre seg.

Oppgavetekst

Skriv en funksjon **moveTo** i **Department** som tar inn en avdeling, og flytter denne avdelingen til å ligge under den oppgitte avdelingen.
*Tips: Du kan trenge en **removeDepartment** metode.*

Testing

Kjør JUnit testen som finnes i filen oppgave3-test.txt.

NTNU

Oppgave 4

7 min

Motivasjon

Et viktig poeng med testing er å sjekke at alle såkalte *edge-cases* er håndtert.

Oppgavetekst

Skriv en JUnit test for å sjekke at **moveTo** ikke tillater at to avdelinger kan være underavdelinger av hverandre. Fiks eventuelle feil som oppdages.

Testing

Kjør JUnit testen du har skrevet, og sjekk at den kjører riktig.

Oppgave 5

5 min

Motivasjon

Vi har nå implementert og testet mesteparten av funksjonaliteten rundt avdelinger og tilhørighet blant avdelinger. Vi er derfor klare til å begynne på ansatte.

Oppgavetekst

Lag en klasse **Employee** og tilhørende konstruktør. En ansatt skal vite hvilken avdeling han/hun tilhører og få dette oppgitt ved konstruksjon.

Testing

Ikke veldig relevant.

NTNU

Oppgave 6

7 min

Motivasjon

Det er ikke veldig nyttig at kun den ansatte vet hvilken avdeling han/hun tilhører.

Oppgavetekst

Utvid **Department** klassen med metoden **addEmployee** som tar vare på at en oppgitt ansatt tilhører avdelingen. Husk å kalle metoden i konstruktøren til **Employee**.

Testing

Ikke veldig relevant.

NTNU

Oppgave 7

7 min

Motivasjon

Det er nyttig å vite hvilke ansatte som tilhører en avdeling eller en av underavdelingene dens.

Oppgavetekst

Lag en metode **getEmployees** i **Department** som returnerer en liste over alle ansatte som tilhører denne avdelingen og dens underavdelinger.

Testing

Kjør JUnit testen som finnes i filen oppgave7-test.txt.

Oppgave 8

7 min

Motivasjon

Ansatte burde kunne flyttes mellom avdelinger.

Oppgavetekst

Lag en metode **moveTo** i **Employee** som tar en avdeling og flytter den ansatte ditt.
*Tips: Du kan trenge en **removeEmployee** metode i **Department***

Testing

Se neste oppgave.

NTNU

Oppgave 9

7 min

Motivasjon

På samme måte som for flytting av avdelinger, er det mye som kan gå galt. Derfor ønsker vi å skrive tester som sjekker at ting fungerer som det skal.

Oppgavetekst

Lag en klasse **EmployeeTest** og skriv en test som sjekker at **moveTo** flytter ansatte riktig.

Testing

Kjør JUnit testen du nettopp skrev, og sjekk at den kjører riktig.

NTNU

Oppgave 10

5 min

Motivasjon

Ansatte kan få forfremmelser. De vil da bli flyttet til avdelingen over.

Oppgavetekst

Lag en metode **promote** i **Employee** som forfremmer den ansatte hvis mulig.

Testing

Kjør JUnit testen som finnes i filen oppgave10-test.txt