



NTNU

Kunnskap for en bedre verden

Kollokvium 4 - Oppgaver

TDT4100 Objektorientert programmering

Mandag 09.03.2020



Oppgaver

I dette kollokviet skal vi jobbe videre med grensesnitt og se på streams. Vi kommer til å fortsett å jobbe med koden vi skrev forrige øvingsforelesning.



Pakker

Utdelt kode og oppgavene finnes i mappen foreksempel/src/kollokvie4.

`foreksempel/src/kollokvie4.kode`

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

`foreksempel/src/kollokvie4.underveis`

Her blir kode lagt inn rett etter felles gjennomgang.

`foreksempel/src/kollokvie4.testar`

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

`foreksempel/src/kollokvie4.lf`

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

Oppgave 1



10 min

Motivasjon

Vi har en **FarmIterator** for å iterere over alle dyrene på en gård. Nå ønsker vi muligheten til å kunne iterere over kun de dyrene som er under 2 år.

Oppgavetekst

Lag en klasse **YoungAnimalsIterator** som tar inn et **Farm** objekt og implementerer **Iterator<Animal>**. Klassen skal bare gi ut de dyrene på gården som er under 2 år.

Testing

main-metoden i filen *oppgave1-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal kun to av dyrene skrives ut.

Oppgave 2

 7 min

Motivasjon

Hvis vi nå ønsker å filtrere på noe annet, må vi skrive en helt ny iterator. Dette er tungvint hvis vi gjøre mange forskjellige filtreringer.

Oppgavetekst

Lag en klasse **FilterAnimalsIterator** som fungerer på samme måte som **YoungAnimalsIterator**, med unntak av at den bruker en implementasjon av grensesnittet **Predicate** til filtrering. Lag også en klasse **YoungAnimalPredicate** som implementerer **Predicate** og sjekker om dyret er under 2 år.

Testing

main-metoden i filen *oppgave2-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal det samme som i oppgave 1 skrives ut.

Oppgave 3

 5 min

Motivasjon

Vi kan nå utnytte **FilterAnimalsIterator** til å filtrere på andre ting enn kun alder.

Oppgavetekst

Lag en klasse **DogPredicate** som implementerer **Predicate** og tester om dyret er av typen hund.

Testing

main-metoden i filen *oppgave3-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal kun de to hundene skrives ut.

Oppgave 4

 7 min

Motivasjon

Vi lagde tidligere **Comparator** objekter for å sammenligne to dyr basert på alderen deres. Vi ønsker nå å sortere en liste av dyr basert på alfabetisk rekkefølge av navnene deres.

Oppgavetekst

I filen *oppgave4-main.txt* finnes det en **main**-metode som lager en liste med dyr, sorterer disse på alder og skriver dem ut i sortert rekkefølge. Endre denne til å sortere dyrene i alfabetisk rekkefølge uten å endre på koden utenfor metoden.
Tips: Se på [lambda](#) uttrykk.

Testing

Kjør **main**-metoden.



Oppgave 5

 5 min

Motivasjon

Etter hvert blir dyr eldre, og vi ønsker derfor en måte å gjøre dyr eldre på i grensesnittet vårt.

Oppgavetekst

Utvid **Animal** grensesnittet med en ekstra metode **incrementAge** som øker alderen til dyret med 1.

Testing

main-metoden i filen *oppgave5-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal først to av dyrene skrives ut, og så bare et.

Oppgave 6

 7 min

Motivasjon

Vi ønsker nå en mulighet for å inkrementere alderen til alle dyrene på en gård.

Oppgavetekst

Lag en metode **newYear** i **Farm** klassen som inkrementerer alderen til alle dyrene. Metoden skal verken bruke **for**- eller **while**-løkker, men skal i stedet bruke **streams**.

Testing

main-metoden i filen *oppgave6-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal den skrive ut det samme som i oppgave 5.

Oppgave 7



10 min

Motivasjon

Vi ønsker nå muligheten for å finne navnene til alle dyrene på gården. Vi ønsker ikke å ta vare på duplikate navn.

Oppgavetekst

Lag en metode **getAnimalNames** som returnerer en liste med navnene til dyrene på gården.

Tips: Bruk **streams**, og se gjerne på **map**, **distinct** og **collect**

Testing

main-metoden i filen *oppgave7-main.txt* kan kjøres for å teste implementasjonen din. Ved riktig implementasjon skal den skrive ut tre navn.