DAT108 Oppgaver repetisjon / oppvarming

Lars-Petter Helland, 18.08.2024

Oppgave 1

Denne oppgaven handler om å skrive ut en tabell med heltall på en «formatert» måte, dvs. at hvert tall representeres som en streng. F.eks. kan tabellen med verdiene [2, 3, 1, 5] skrives ut slik:

★ ★
★ ★ ★
★
★ ★ ★ ★ ★

Løsningen skal være slik:

- Det skal lages en metode **`void skrivUtTallene(int[] tabell, Tallformat format)`** som skriver ut alle tallene i **tabell**, formatert ved hjelp av objektet **format**.
- Tallformat er et interface med metoden **`String somStreng(int tall)`**.
- Det må lages en klasse for hver måte å formatere på, f.eks. **`Stjerneformat`** for eksemplet over.
- Test ut i main() f.eks. med eksemplet over.

Oppgave 2

Denne oppgaven ligner litt på den forrige, men har et par ekstra ingredienser. Vi skal gjøre tilsvarende til forrige oppgave, men nå med en **liste av personer**.

Løsningen skal være slik:

- Personklassen skal helst defineres som en **record**, altså **`public record Person(...)`**. (Begynn gjerne å ha det som en class hvis dette blir forvirrende. Det funker, det også)
- Listen av personer i main() skal opprettes med **`List.of(...)`**.
- Tilsvarende som i Oppgave 1 om metode for å skrive ut alle personene i listen.
- Tilsvarende som i Oppgave 1 om interface og klasse for å lage utskriftsstreng for én person.

F.eks. kan personer inneholde **fornavn**, **etternavn**, **fødselsår** og **mobil**. Og vi kan f.eks. tenke oss en utskrift der navnene er med store bokstaver, teksten «født i» er lagt inn, og mobilnummeret er utelatt:

ATLE PATLE, født i 1970
PER VISKELER, født i 1972
MADAM FELLE, født i 1960
DONALD DUCK, født i 1950

Oppgave 3

Denne er litt mer krevende, men dette er også repetisjonsstoff. Den handler om å jobbe med intervaller av verdier.

Vi kan tenke oss en del bruksområder for en **datatype intervall**. F.eks. kan vi ha:

- Et intervall av poeng (int), f.eks. [90, 100]
- Et intervall av temperaturer (double), f.eks. [15.0, 30.0]
- Et intervall av klokkeslett (egendef comparable type), f.eks. [10:15, 12:00]
- Varianter lukket [a,b], halvåpent [a,b> og helåpent <a,b> intervall.

En slik intervalltype kan f.eks brukes til å finne ut:

- **Om en temperatur er i et gitt intervall** (om 12.0 er i intervallet sommertemperaturer ...)
- **Om to møter kolliderer/overlapper** (om møtet [10:15, 12:00> kolliderer med [11:15, 13:00> ...)
- Hva som er **den minste verdien** og **den største verdien** i intervallet.

Deloppgaver Oppgave 3

1. Definer en ADT **Intervall** for et slikt intervall med de 4 metodene som er antydet (uthevet) over.
    - Tips: Start med et intervall av heltall for enkelhets skyld.
    - Tips: Det er nok også enklest å holde seg til én intervall-variant, f.eks. et lukket intervall [a,b].

2. Lag en implementasjon **IntervallImpl** av denne intervall-typen.
    - Tips: Du trenger ikke å lagre alle verdier, kun min og maks.

3. Lag en **enhetstest** som tester at IntervallImpl virker som den skal.
    - Tips: Hva er fornuftig(e) konstruktør(er)?
    - Tips: Hvordan skal den oppføre seg hvis man prøver å lage et ugyldig intervall?

4. Generics. I stedet for å la det være et intervall av heltall, skriv det om til å være **et intervall av T**, der T er en **Comparable<T>,** dvs. at T kan sammenlignes med andre T-er.

5. Skriv om enhetstesten fra pkt.3 til å bruke den nye implementasjonen med **<Integer>** som typeparameter.

6. Legg til et par tester på et intervall av desimaltall **<Double>** (i tillegg til de på heltall).

7. Skriv en egen klasse **Klokkeslett** (time og minutt) og skriv **enheltstester for intervall av <Klokkeslett>**.
    - Tips: Klokkeslett må implementere Comparable<Klokkeslett>.

18. august 2024
Lars-Petter Helland