Trinn 10 – Set, Optional, equals og hashCode

Tenk at vi nærmer oss slutten i kurset. Vi har vært igjennom de største temaene i emnet, så nå kan vi lære oss diverse mindre temaer som man absolutt bør beherske når man lærer Java.

Også denne gangen har jeg laget en lengre intro-video, så se gjerne den før du begynner.

Mål for dette trinnet:

- Jeg behersker bruk av et Set, og jeg vet når det kan være hensiktsmessig.
- Jeg skjønner hvordan jeg kan benytte Optional.
- Jeg vet når det er lurt å implementere equals og hashCode for objekter.
- Jeg skjønner behovet for å validere input i metoder.

Relevante kapitler i pensumbokas:

- Kapittel 17: Comparing objects
- Kapittel 22: Set

Her er spørreskjemaet for trinn 10 der du kan fortelle hvordan det gikk.

Vi skal bli kjent med en ny type collection, nemlig Set. Vi skal få trening i å legge inn og hente ut objekter i et Set. Videre skal vi se på bruken av Optional, hvordan vi kan validere input til en metode og hvorfor equals/hashCode kan være lurt å implementere.

Oppgave 1 – Lage utgangspunkt

Opprett et nytt prosjekt. Lag en klasse med en main-metode. Lag en klasse Program. I main-metoden, opprett et objekt av klassen Program.

Lag en klasse Person. Klassen skal ha to private fields: age (int) og name (String). Lag gettere og settere og en toString-metode. Lag en konstruktør som tar imot både age og name (og setter verdiene i fieldsene).

Hvis du sliter med denne oppgaven, så kan du velge å benytte kode-utgangpunktet som ligger i Canvas. Kjør i så fall programmet for å se om importeringen av koden gikk bra. Forventet resultat: Person{age=20, name='Anton Antonsen'}

Oppgave 2

Endre Program-klassen slik at den har et HashSet som kan holde i Person-objekter.

Lag en metode i Program. Metoden skal

- Opprette to objekter av klassen Person. Begge personer skal ha **samme verdier** for navn og alder (for eksempel Atle Antonsen, 20 år).
- Putt begge objektene i hash-settet.
- Skrive ut alle verdiene i hash-settet (etter at objektene ble lagt inn).

Kall metoden fra main. Hva ble resultatet? Er det ikke slik at et sett bare skal holde unike verdier? Tenk litt på det😊

Oppgave 3

Lag equals- og hashCode-metoder i klassen Person. Disse kan du autogenerere i IntelliJ (vist i introvideoen). Kjør programmet på nytt. Hva skjedde? Forhåpentligvis har du sett at innholdet i hashsettet endret seg i forhold til forrige kjøring. Tenk igjennom hvorfor endringen skjedde.

Oppgave 4

Lag en metode getSamplePerson(String name) i Program-klassen. Metoden skal hente ut en person i hashsettet (hvis det finnes en person der som matcher på navn). Hvis det er flere personer i hash-settet med navnet, så er det ikke viktig hvilken av disse som returneres. Hvis personen ikke finnes i hash-settet, så skal metoden **ikke** returnere null (sjekk intro-videoen hvis du er usikker på hva som menes med det).

Test at metoden fungerer som tiltenkt ved å forsøke å hente ut en person som finnes, og som ikke finnes i hash-settet.

Oppgave 5

Lag en metode addPerson (Person p) som legger en person inn i hash-settet. Du skal validere input til metoden. En person skal ikke kunne være *null*, og alderen til en person kan ikke være negativ. Navnet til personen kan ikke være *null*.

Sjekk om valideringen fungerer ved å forsøke å benytte addPerson til å legge til gyldige og ikke gyldige personer.

Oppgave 6

Ved bruk av metoden *addPerson*, legg inn ti personer inn i hash-settet.

Lag deretter en metode som tar imot en *alder* (int) og returnerer et Set med personer (fra hash-settet) som er eldre enn alderen *alder*.

Oppgave 7

Gjør deg bedre kjent med HashSet ved å forsøke ut ulike metoden som klassen tilbyr. Bruk javadokumentasjonen for Java 15 til å finne hvilke metoder som klassen tilbyr.

Ekstraoppgave

I dette emnet kommer vi ikke inn på funksjonelle interface. Men du kan ta en titt her allikevel 😊
Dette er altså ikke eksamens-relevant, men nyttig å vite når man programmerer i Java.