Ukesoppgaver, gruppetime 10 - Invarianter

Oppgave 1: Løkkeinvariant og før- og etterbetingelser (Handlekurv)

a) Definér løkkeinvarianten forbundet med Handlekurv-klassen nedenfor.

```
public class Handlekurv {
    double[] produktpriser;
    public Handlekurv(double[] produktpriser) {
        this.produktpriser = produktpriser;
    }

    public double totalPris() {
        double total = 0.0; int i = 0;
        while(i < produktpriser.length) {
            total += produktpriser[i];
            i++;
        }
        return total;
    }
}
```

b) Hva er naturlige før- og etterbetingelser forbundet med Handlekurv-klassen ovenfor?

Oppgave 2: Klasseinvarianter og før- og etterbetingelser (Bankkonto)

a) Definér klasseinvariantene forbundet med Bankkonto-klassen nedenfor.

```
public class Bankkonto {
    private String kontonummer; private double saldo;

    public Bankkonto(String kontonummer) {
        if (kontonummer == null || kontonummer.isEmpty()) {
            throw new IllegalArgumentException("[Passende feilmelding]");
        }
        this.kontonummer = kontonummer; this.saldo = 0.0;
    }

    public void innskudd(double belop) {
        if (belop <= 0) {
            throw new IllegalArgumentException("[Passende feilmelding]");
        }
        this.saldo += belop;
    }
```

```
    public void uttak(double belop) {
        if (belop <= 0) {
            throw new IllegalArgumentException("[Passende feilmelding]");
        }
        if (this.saldo < belop) {
            throw new IllegalArgumentException("[Passende feilmelding]");
        }
        this.saldo -= belop;
    }
}
```

b) Hva er naturlige før- og etterbetingelser forbundet med Bankkonto-klassen ovenfor?

Oppgave 3: Programmering basert på en løkkeinvariant (Fruktbutikk)

Tenk deg at du er eier av en fruktbutikk, og du vil sjekke om du har nok frukt på lager. Du har et HashMap over behovet for hver frukttype og et annet HashMap over lageret ditt. Du får oppgitt følgende løkkeinvariant:

- For alle frukttyper, er det sant at for alle tidligere frukttyper, tilsvarer antall frukt av typen x på lager minst antallet frukt av typen x det er behov for.

Oppgaven din er å implementere en klasse med metoder som ivaretar løkkeinvarianten for fruktbutikken ovenfor. Du skal bruke rekursjon.

Oppgave 4: Programmering basert på klasseinvarianter (Leilighetskompleks)

Du får oppgitt følgende klasseinvarianter tilknyttet et leilighetskompleks:

- Antall leiligheter per etasje: Hver etasje skal alltid ha et array med plass til nøyaktig 15 leiligheter.
- Datatypen til elementene: Hvert element i arrayet av leiligheter er et Leilighet-objekt.
- Gyldige etasjenumre: Nøklene i HashMap (etasjenummer) skal være strenger på formatet "etasje1", "etasje2", o.s.v.
- Unike nøkler: Hver nøkkel i HashMap (etasjenummer) skal være unik.
- Ikke null-nøkler: Ingen av nøklene (etasjenumrene) skal være null.
- Gyldige verdier: Verdiene (arrays av leiligheter) må alltid være gyldige arrays av riktig størrelse (i dette tilfellet 15).
- Ikke null-verdier: Ingen av verdiene i HashMap skal være null.

Oppgaven din er å implementere klasser og metoder som ivaretar klasseinvariantene for leilighetskomplekset definert ovenfor.

Oppgave 5: Parprogrammering med hensyn på invarianter (Produktkatalog)

Parprogrammering fungerer slik at én av dere beskriver hva som skal implementeres og én av dere programmerer av gangen. Bytt gjerne roller underveis. Dere har fått i oppdrag å lage en Produktkatalog, der produktene lagres i et array med plass til 20 produkter. Til oppgaven

trenger dere en Produkt-klasse som skal holde på en produktID (int), et produktnavn, en produktpris og en produktbeskrivelse. For å legge til et produkt trengs en leggTil-metode, som legger til på neste ledige plass i arrayet. I tillegg trengs en fjern-metode, som fjerner et produkt basert på produktID. Ved fjerning skal de gjenværende produktene til høyre for det fjernede flyttes ett hakk til venstre. Da unngår dere null-verdier mellom to produkter i arrayet. Merk at det kan være nyttig med en antall-variabel i produktkatalogen. En del av oppgaven er å definere klasseinvarianter og løkkeinvarianter for Produktkatalog i samarbeid, før dere starter på selve parprogrammeringen som skal hensynta invariantene.