

# INF1010 - Seminaroppgaver til uke 7

## Oppgave 1

Sett inn riktig parameter, slik at metoden tar inn en Collection som holder på klassen Person *eller* en subklasse av denne.

```
public      void printPersoner(                )
```

## Oppgave 2

Anta at vi har laget følgende lister. Du kan anta at listene settes tilbake sin opprinnelige form etter hver deloppgave.

```
Collection<String> listeA = new ArrayList<String>();  
listeA.add("INF1010");  
listeA.add("INF1510");  
listeA.add("INF1050");
```

```
Collection<String> listeB = new HashSet<String>();  
listeB.add("INF1010");  
listeB.add("INF1411");  
listeB.add("MAT1100");
```

a) Hva er resultatet i listeA etter følgende kjøring?

```
listeA.addAll(listeB);
```

b) Hva er resultatet i listeA etter følgende kjøring?

```
listeA.retainAll(listeB);
```

c) Hva er resultatet i `listeA` etter følgende kjøring?

```
listeA.removeAll(listeB);
```

d) Hva er resultatet i `boolean delmengde`?

```
boolean delmengde = listeA.containsAll(listeB);
```

### Oppgave 3

Skriv en metode som tar inn to `Collections`, som skal legge til alle elementene i det andre parameteret til det første parameteret. Det er et krav at det første parameteret holder på en liste av samme type eller en superklasse av det andre parameteret. Hvis vi antar at vi har en `Collection<Object> objektsliste` og `Collection<Student> studentliste`, vil `leggSammenLister(objektsliste, studentliste)` være godkjent, men `leggSammenLister(objektsliste, studentliste)` vil feile. Metoden trenger ikke å ha noen returverdi.

### Oppgave 4

Hvilken funksjon har tegnet ? når vi holder på med parametriserte klasser?

### Oppgave 5

Anta at du har en `Collection<Integer> tall`.

a) Vis hvordan du kan gå igjennom samlingen ved hjelp av en `for-each` løkke.

- b) Vis hvordan du kan gå igjennom samlingen ved hjelp av en iterator.

## Oppgave 6

Hva er forskjellen mellom grensesnittet `Iterable` og `Iterator`?

## Oppgave 7

Fyll inn parametre slik at klassen kun tar inn objekter av en klasse som enten er klassen `Person` eller en subklasse av denne.

```
public class ListeAvPersoner<                >
```

## Oppgave 8

Fyll inn parametre slik at klassen kun tar inn objekter av en klasse som implementerer grensesnittet `Comparable`.

```
public class ListeAvSammenlignbareObjekter<                >
```

## Oppgave 9

Lag en klasse `SimpleArrayList` som er en liste basert på et array. Krav til listen:

- Listen skal inneholde et array, og elementene i listen skal puttes på neste ledige plass i arrayen når de blir lagt til.
- For enkelhets skyld vil vi ikke tillate sletting fra listen, kun innlegging.
- Listen skal ha en bestemt kapasitet som konstruktøren tar som parameter. Du trenger ikke implementere utvidelse av kapasiteten.
- Listen skal være itererbar, dvs. implementere grensesnittet `java.util.Iterable` med medfølgende implementasjon av `java.util.Iterator`.

(Hint: Generiske arrayer er tildels vanskelige å jobbe med. Bruk heller et `Object-array`.)