

# INF1000 - Uke 10

## Løsningsforslag ukesoppgaver 10

24. oktober 2012

### Løsningsforlag

#### Oppgave 1

##### Array vs. HashMap

a) Følgende program viser et enkelt banksystem med en array `kontoer[]`, og metoder for å finne en konto vha. navn til eieren og vha. kontonummer. Skriv om programmet slik at det bruker en `HashMap` i stedet for arrayen `kontoer[]`. I første omgang lager vi én `HashMap`, med personnavn som nøkkel og et `Konto`-objekt som verdi, deklartert slik:

```
HashMap<String, Konto> kontoFraNavn = new HashMap<String, Konto>();
```

Hvilke fordeler og ulemper får vi av å bruke `HashMap` her? Hva kan variabelen `antKontoer` erstattes med i programmet? (Anta foreløpig at personnavnene er unike og at hver person bare kan ha én konto i banken.)

```
class Konto {
    int nr; // kontonummer
    String navn; // eier
    int saldo;

    Konto(int nr, String navn, int saldo) {
        this.nr = nr;
        this.navn = navn;
        this.saldo = saldo;
    }

    void settInn(int innskudd) {
        saldo = saldo + innskudd;
    }
}

class Bank {
    Konto[] kontoer = new Konto[1000];
    int antKontoer = 0;

    public static void main(String[] args) {
        Bank b = new Bank();
    }
}
```

```

Bank() {
    åpneNyttKonto(530010, "Nils", 4000);
    åpneNyttKonto(720020, "Elin", 8000);
    åpneNyttKonto(910030, "Tina", 9000);

    Konto k = finnKontoFraNavn("Elin");
    System.out.println("Elins kontonr: " + k.nr + ", saldo: " + k.saldo);

    k = finnKontoFraNr(530010);
    System.out.println("Kontonr. " + k.nr + " tilhører " + k.navn);
}

void åpneNyttKonto(int nr, String navn, int saldo) {
    Konto k = new Konto(nr, navn, saldo);
    kontoer[antKontoer] = k;
    antKontoer++;
}

Konto finnKontoFraNavn(String navn) {
    for (int i = 0; i < antKontoer; i++) {
        if (kontoer[i].navn.equals(navn)) {
            return kontoer[i];
        }
    }
    return null;
}

Konto finnKontoFraNr(int kontonr) {
    for (int i = 0; i < antKontoer; i++) {
        if (kontoer[i].nr == kontonr) {
            return kontoer[i];
        }
    }
    return null;
}
}

```

```

KJØREEKSEMPEL:
Elins kontonr: 720020, saldo: 8000
Kontonr. 530010 tilhører Nils

```

## Løsningsforslag

```

import java.util.*;

class Konto {
    int nr; // kontonummer
    String navn; // eier
    int saldo;

    Konto(int nr, String navn, int saldo) {
        this.nr = nr;
        this.navn = navn;
    }
}

```

```

        this.saldo = saldo;
    }

    void settInn(int innskudd) {
        saldo = saldo + innskudd;
    }
}

class Bank2 {
    HashMap <String, Konto> kontoFraNavn = new HashMap <String, Konto>();
    HashMap <String, Konto> kontoer = new HashMap <String, Konto>();

    public static void main(String[] args) {
        Bank2 b = new Bank2();
    }

    Bank2() {
        åpneNyttKonto(530010, "Nils", 4000);
        åpneNyttKonto(720020, "Elin", 8000);
        åpneNyttKonto(910030, "Tina", 9000);

        Konto k = finnKontoFraNavn("Elin");
        System.out.println("Elins kontonr: " + k.nr + ", saldo: " + k.saldo);

        k = finnKontoFraNr(530010);
        System.out.println("Kontonr. " + k.nr + " tilhører " + k.navn);

        System.out.println("Antall kontoer: " + kontoer.size());
        avsluttKonto(k);
        System.out.println("Fins kontoen nå?: " + kontoer.containsValue(k));
        System.out.println("Antall kontoer nå: " + kontoer.size());
    }

    void åpneNyttKonto(int nr, String navn, int saldo) {
        Konto k = new Konto(nr, navn, saldo);
        kontoFraNavn.put(navn, k);
        kontoer.put("" + nr, k);
    }

    Konto finnKontoFraNavn(String navn) {
        return kontoFraNavn.get(navn);
    }

    Konto finnKontoFraNr(int kontonr) {
        return kontoer.get("" + kontonr);
    }

    void avsluttKonto(Konto k) {
        kontoFraNavn.remove("" + k.nr);
        kontoer.remove("" + k.nr);
    }
}

```

KJØREEKSEMPEL:

Elins kontonr: 720020, saldo: 8000  
Kontonr. 530010 tilhører Nils  
Antall kontoer: 3  
Fins kontoen nå?: false  
Antall kontoer nå: 2

**b)** Lag en HashMap til, kalt kontoer, hvor du bruker som nøkkel konto-nummeret konvertert til String, og fortsatt Konto-objektene som verdi. Vis at metoden finnKontoFraNr() blir enklere nå. Videre tenk deg at vi skal ha en metode for å fjerne en konto. Følgende kode viser hvordan det kan gjøres med arrayer. Hvor mange programsetninger trengs det når vi bruker én HashMap i stedet? Og med to?

```
void avsluttKonto(Konto k) {
    // Fjerner en konto ved å finne indeksen til kontoen i arrayen
    // kontoer[] og flytte alle kontoene med høyere indeks en plass ned.
    boolean funnet = false;
    for (int i = 0; i < antKontoer && !funnet; i++) {
        if (kontoer[i] == k) {
            funnet = true;
            for (int j = i; j < antKontoer - 1; j++) {
                kontoer[j] = kontoer[j + 1];
            }
            antKontoer--;
        }
    }
}
```

## Løsningsforslag

Se del a).

**c)** Disse oppgavene har begrensningen at personnavnene må være unike og at hver person bare kan ha én konto i banken. Hvordan ville man unngått disse begrensninger i et mer avansert system? Hvilke fordeler og ulemper ser du av å bruke HashMap-er i stedet for 2D-arrayer i Oblig 3? (foreslå mulige nøkkel/verdi-kombinasjoner).

**Hint:** Se avsnitt 9.11 på side 191 i læreboka for forskjellene mellom arrayer og HashMap-er.

## Løsningsforlag

I store systemer passer det bedre å ha personnummer som nøkkel eller noe annet som er unikt for hver verdi, i stedet for navn alene. Verdi kan være et Person-objekt, som i sin tur kan ha en objektvariabel av type HashMap som refererer til alle kontoene for personen. Hvis programmet trenger å finne personer ut fra navn så kan man likevel bruke navn som nøkkel i en tilleggs-HashMap, som f.eks. kan ha som verdi en HashMap over personene med det navnet.

**Arrayer** passer best å bruke når maks. antall elementer er kjent på forhånd, når vi trenger en gitt rekkefølge på dataene, og når indeksene er numeriske.

**HashMap** er bedre når maks. antall elementer er ukjent, eller man trenger å slå opp i dataene vha. "indeksersom er noe annet enn tall (f.eks. tekster), og når rekkefølgen på dataene ikke er avgjørende. Med HashMap må dataene være objekter (pekere), ikke primitive typer, men det finnes objekt-typer for alle de primitive: Integer, Double, Character, Boolean.

I **Oblig 3** er det mest naturlig å bruke arrayer, men man kan også bruke HashMap: da kan f.eks. hele hybelnavnet være nøkkel. Man trenger likevel lett tilgang til etasje- og rom-nr. så disse kan evt. legges til som objektvariabler i Hybel. En annen fremgangsmåte er å ha Etasje som egen klasse og da kan denne ha en array (eller HashMap) med hyblene i etasjen.

### Oppgave 3

#### Innstikksortering av kontoer:

Ta utgangspunkt i følgende program (som sorterer en heltalls-array og en String-array); og lag en metode som sorterer arrayen `kontoer[]` fra oppgave nr. 2 (a) ovenfor alfabetisk på personnavn. Etter et kall på metoden skal f.eks. `kontoer[0]` være Elins konto, `kontoer[1]` Nils sin, osv. Metoden skal ha to innparametre: `Konto[] kontoer`, og `int antKontoer`; og skal kunne sortere et vilkårlig antall kontoer. Utvid programmet i nr. 2 (a) med et kall på metoden for å sjekke at den fungerer.

```
class Innstikksort {

    /** Sorterer en heltallsarray. */
    public static void sorter(int[] a) {
        for (int k = 0 ; k < a.length - 1; k++) {

            if (a[k] > a[k + 1]) {
                // a[k + 1] står på feil plass, ta den ut:
                int tmp = a[k + 1];
                int i = k;

                // Skyv a[i] mot høyre ett hakk til vi finner
                // riktig plass til tmp:
                while (i >= 0 && a[i] > tmp) {
                    a[i + 1] = a[i];
                    i--;
                }
                // Sett tmp inn på riktig plass:
                a[i + 1] = tmp;
            }
        }
    }

    /** Sorterer en String-array. Dette er en redigert utgave av
    metoden ovenfor som jobber med String-array i stedet for int-array. */
    public static void sorter(String[] a) {
        for (int k = 0 ; k < a.length - 1; k++) {
            if (a[k].compareTo(a[k + 1]) > 0 ) {
                String tmp = a[k + 1];
```



```
        kontoer[i + 1] = tmp;
    }
}
}
```

Legger også til følgende testkode på slutten av konstruktøren til Bank:

```
// Kontosortering:
System.out.println("Sorterer kontoene på navn...");
sorter(kontoer, antKontoer);
for (int i = 0; i < antKontoer; i++) {
    System.out.println("kontoer[" + i + "].navn = " + kontoer[i].navn
        + ", saldo: " + kontoer[i].saldo);
}
```

KJØREEKSEMPEL:

```
Sorterer kontoene på navn...
kontoer[0].navn = Elin, saldo: 8000
kontoer[1].navn = Nils, saldo: 4000
kontoer[2].navn = Tina, saldo: 9000
```