

# INF1000 - Uke 10

## Ukesoppgaver 10

24. oktober 2012

### Vanlige ukesoppgaver

De første 4 oppgavene (Oppgave 1 - 4) handler om HashMap og bør absolutt gjøres før du starter på Oblig 4. Deretter er det en del repetisjonsoppgaver som dekker de ulike delene av pensum.

Nøkkeloppgaver er oppgaver merket med [**Nøkkeloppg.**] og er plukket ut som spesielt representativ for de viktigste temaene fra ukens forelesning, og alle bør ha som minimumsmål å løse denne selvstendig.

### Oppgave 1

#### Legg inn i et hashmap:

Du har klassen Dyr (se under). Lag en klasse DyreMap, som inneholder en main-metode. Her skal du deklare et HashMap<String, Dyr>. Opprett minst tre objekter av klassen Dyr, finn på navn og idnr. Husk at idnr bør være unike! Legg disse inn i hashmappet med idnr som nøkkel.

```
class Dyr {
    String navn;
    String idnr;

    Dyr(String navn, String idnr) {
        this.navn = navn;
        this.idnr = idnr;
    }

    String getIDnr() {
        return idnr;
    }
}
```

### Oppgave 2 [Nøkkeloppg.]

#### 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 en HashMap, med personnavn som nøkkel og et Konto-objekt som verdi, deklarerert 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 en 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

b) Lag en HashMap til, kalt kontoer, hvor du bruker som nøkkel kontonummet 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 en 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--;
        }
    }
}
}

```

c) Disse oppgavene har begrensningen at personnavnene må være unike og at hver person bare kan ha en 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-arryaer 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.

### Oppgave 3

#### HashMap: Hva skrives ut?

```
import java.util.*;
class Personer {
    public static void main(String[] args) {
        HashMap register = new HashMap ();

        Person p1 = new Person("Ida", 19);
        Person p2 = new Person("Lars", 21);

        register.put(p1.navn, p1);
        register.put(p2.navn, p2);

// a)
        Person p = register.get("Ida");
        System.out.println(p.navn + p.alder);

// b)
        for (String s : register.keySet()) {
            System.out.println(s);
        }

// c)
        p1.alder = 24;
        for (Person p3 : register.values()) {
            System.out.println(p3.navn + ":" + p3.alder);
        }

// d)
        if (register.containsValue(p2) && !register.containsKey("Elin")) {
            System.out.println(true);
        }

// e)
        register.remove("Lars");
        System.out.println(register.size() + " - " + register.isEmpty());

// f)
        System.out.println(register.remove("Ida") == null);
        System.out.println(register.remove("Ida") == null);
    }
}

class Person {
    String navn;
    int alder;

    Person(String navn, int alder) {
        this.navn = navn;
        this.alder = alder;
    }
}
```

## Oppgave 4 [Nøkkeloppg.]

### HashMap med akronymer: kap. 9, oppg. 2 (noe modifisert tekst) (side 195)

a) Et akronym er et initialord (forkortelse) som blir lest og uttalt som et vanlig ord, altså ikke bokstav for bokstav. Lag et program som leser inn en rekke akronymer og deres tolkning fra fil (du finner en fil med akronymer (og andre forkortelser) på bokas hjemmeside: akronymer.txt). Legg akronymene og tolkningene i en HashMap med akronymet som nøkkel.

Brukeren skal oppgi et akronym til programmet og få tilbake en eller flere tolkninger av akronymet. Det kan maksimalt være 10 tolkninger til hvert akronym. Denne brukerdialogen skal gå i løkke til brukeren svarer ".

**Tips:** Lag en klasse der objekter har følgende attributter: akronymet og en String-array med tolkninger, samt antall tolkninger. Pekere til objekter av denne klassen legges i HashMap med akronymet som nøkkel. Det kan være lurt å lage en metode i denne klassen som legger til en ny tolkning, og som holder orden på hvor mange tolkninger det er i øyeblikket.

```
API  Application Programming Interface
AV   Audio/Video
AV   Authenticity Verification
BASH Bourne Again Shell [Unix]
EU   Europaunionen
FN   Forente nasjoner
... osv ...
```

b) Utvid programmet slik at bruker får anledning til å legge til en tolkning dersom den ikke finnes fra før.

## Oppgave 5 [Nøkkeloppg.]

### Uke 3: Array med tall

a) Lag et program med en for-løkke som ber bruker taste inn fem heltall og lagrer disse i en array kalt tall:

```
int[] tall = new int[5];
```

b) **Sum av array:** Utvid programmet slik at det regner ut summen av tallene ved hjelp av en løkke, og skriver ut resultatet.

c) **Minste verdi:** Utvid programmet slik at det finner og skriver ut det minste tallet i arrayen.

d) **Lave verdier:** Legg til programkode som skriver ut alle verdiene i arrayen som er mindre enn 10.

e) **Søk:** Legg til programkode som skriver ut en beskjed om verdien 5 finnes eller ikke finnes i arrayen.

## Oppgave 6

### Metoder (10 poeng) oppgave 5 fra eksamen h09

a) Nedenfor er det deklart en metode `sum(int[] array)`, som skal beregne summen av tallene i en int-array. Vil metoden compilere? Hvis nei, forklar hva som må endres for at metoden skal compilere.

```
public void sum ( int[] array ) {  
    int sum = 0;  
    for (int i=0; i < array.length: i++) {  
        sum += array { i };  
    }  
    return sum;  
}
```

b) Deklarer en metode `snitt(int[] array)`, som skal beregne gjennomsnittet av tallene i parameteren array av type int-array. Snittet skal beregnes med flyttallsdivisjon og returverdien skal være av type double. Du skal skrive hele metoden, inkludert signaturen.

c) Vi kan beregne en gjenstands snitthastighet ved å dividere tilbakeløst distanse på tid. Deklarer en metode som tar to double-parametre distanse og tid, og som returnerer hastigheten som en double-verdi. Hvis verdien til tid er 0, skal metoden returnere -1. Du skal skrive hele metoden, inkludert signaturen. Finn på et passende metodenavn.

d) I sjø- og luftfart oppgis ofte hastigheter i knop. Én knop er definert som én nautisk mil pr. time, som tilsvarer 1852 meter pr. time. Deklarer en metode som tar en double-parameter knop og returnerer den tilsvarende hastigheten i km/t som en double-verdi. Du skal skrive hele metoden, inkludert signaturen. Finn på et passende metodenavn.

## Oppgave 7

### Klasser og objekter, Oppgave 11 eksamen h05 (25 poeng)

I programmet nedenfor skal du lage en konstruktør til klassen Pyramide som har pyramidens bredde, lengde og høyde som tre double-parametreparameter. Du skal også lage en objektmetode i klassen Pyramide som regner ut volumet og returnerer denne verdien (du skal bruke formelen:  $\text{volum} = 0.333 \cdot \text{høyde} \cdot \text{bredde} \cdot \text{lengde}$ ). Du skal også skrive programkode i main som oppretter to

pyramider, en med langde =30.1,bredde=30.1 og høyde = 22.9 og en med samme grunnflate, men med dobbelt så stor høyde som den første pyramiden. Fra main skal du så kalle på volumberegningemetoden i hvert av de to objektene og skrive ut en linje for hver pyramide med høyde, lengde og bredde samt volumet.

```
class PyramideTest {  
  
    public static void main ( String [] args) {  
        // skriv kode her som lager to Pyramide-objekter og  
        // skriver ut deres høyde, lengde og volum  
  
    } // end main  
} // end class PyramideTest  
  
class Pyramide {  
    double høyde, lengde, bredde;  
    // skriv konstruktør her  
  
    // skriv objektmetode her som beregner og returnerer volumet  
  
} // end class Pyramide
```

## Ekstraoppgaver

### Oppgave 1

#### Bytte verdiene i to variable

Lag et program som inneholder to heltallsvariable,  $a = 2$  og  $b = 10$ . Skriv kode som bytter om verdiene i de to variablene, og skriver resultatet til skjerm. (Resultatet skal altså være  $a = 10$  og  $b = 2$ .) Programmet skal fungere for alle verdier av  $a$  og  $b$ , du kan altså ikke skrive  $a = 10$ ; og  $b = 2$ ; direkte i programmet ditt. Prøv med andre verdier for  $a$  og  $b$ , og sjekk at programmet fortsatt fungerer.

### Oppgave 2

#### Løkker: Hva blir skrevet ut?

Avgjør uten å bruke datamaskin hva som blir skrevet ut når følgende programsetninger utføres.

```

// (a)
int a = 10;
while (a < 20) {
    a += 4;
}
System.out.println("a = " + a);

// (b)
int sum = 0;
for (int b = 1; b < 6; b += 2) {
    sum += b;
}
System.out.println("sum = " + sum);

// (c)
int produkt = 1;
for (int c = 1; c < 4; c++) {
    produkt = produkt * c;
    System.out.println(produkt);
}

// (d)
for (int d = 3; d >= 1; d--) {
    for (int e = 1; e <= 3; e++) {
        System.out.println(d + e);
    }
}

// (e)
int teller = 0;
for (int ytre = 0; ytre < 3; ytre++) {
    teller++;
    for (int indre = 0; indre < 3; indre++) {
        teller++;
    }
}
System.out.println(teller);

```