GRUPPETIME UKE 2

IN1010 Gruppe 3

agneshb@uio.no

I dag:

- public, private, protected
- enhetstesting
- array
- ArrayList
- HashMap
- Oppgaver

Innlevering 1

- Frist: 12. februar
- Fylle inn egenmeldingsskjema ved behov for utsettelse
- Obligatorisk!!

Public, private, protected

- **Public**: alle klasser kan se.
- **Protected**: alle subklasser kan se.
- **Private**: ingen klasser eller subklasser kan se, kun klassen selv.

Enhetstesting

- Test hver del eller hvert objekt for seg
- Etter å ha testet hver del kan man teste det sammen (=**integrasjonstesting**)
- Prøv flere ulike rekkefølger på kall.
- Spesielt viktig å sjekke edge-cases:
  - Hva skjer hvis man prøver å hente ut noe fra en tom liste?
  - Hva skjer hvis en variabel har verdi null?
  - Hva skjer hvis man prøver å sette inn noe i et fullt array?

```java
public class Teller {

    int tall;

    public Teller(int starttall){
        tall = starttall;
    }

    void tellOpp(){
        tall ++;
    }

    void tellNed(){
        tall--;
    }

    void hentTeller(){
        System.out.println(tall);
    }
```

```java
    Run | Debug
    public static void main(String[] args) {
        //en teller som starter på 0:
        Teller t = new Teller(starttall:0);

        //sjekker verdien før metoden:
        t.hentTeller();

        //kaller metode som skal endre t sitt tall:
        t.tellOpp();

        //sjekker om verdien er den forventede verdien:

        t.hentTeller();

    }
}
```

Array

Array i Java

- **Fastsatt størrelse**: kan ikke endre størrelsen etter den er opprettet.
- **Lik type**: Alle elementer i listen må være av samme type.
  - Eks: [1, 3, 5]
- **Ingen metoder**: har ingen metoder knyttet til seg – må gjøre endringer «for hånd».

List i Python

- Dynamisk størrelse: kan alltid legge ti/fjerne flere elementer.
- Ulike typer: listen kan ikkeholde elementer av ulike typer.
  - Eks: [«IN1010», 10, true]
- Tilbyr metoder.

Array

- tallArray: har plass til 4 int-er.
- strArray: har plass til 2 String-er.
- Kan ikke legge til flere elementer enn hva vi setter som størrelse ved opprettelsen.
  - Løsning: lage en ny array med flere plasser og kopiere over ett og et element.

```
int[] tallArray = new int[4];
tallArray[0] = 8;
tallArray[1] = 30;
tallArray[2] = 2;
tallArray[3] = 7;
```

```
String[] strArray = new String[2];
strArray[0] = "IN1010";
strArray[1] = "IN1050";
```

ArrayList

- Klasse i Java-biblioteket
- Størrelsen må ikke oppgis ved opprettelse.
- Størrelsen kan endres ved behov.
- Noen metoder:
  - **size**()
  - **isEmpty**()
  - **contains**(element)
  - **get**(i)
  - **set**(i, element)
  - **add**(element)/**add**(i, element)

```java
public static void main(String[] args) {
    ArrayList<Integer> tall = new ArrayList<>();
    tall.add(e:0);
    tall.add(e:1);
    tall.add(e:3);

    ArrayList<Katt> katter = new ArrayList<>();
    katter.add(new Katt(navn:"Ole"));
    katter.add(new Katt(navn:"Dole"));
    katter.add(new Katt(navn:"Doffen"));

}
```

For each med ArrayList

```java
ArrayList<Katt> katter = new ArrayList<>();
katter.add(new Katt(navn:"Ole"));
katter.add(new Katt(navn:"Dole"));
katter.add(new Katt(navn:"Doffen"));

for (Katt katt : katter) {
    System.out.println(katt.navn);
}
```

```
Ole
Dole
Doffen
```

HashMap

- Som Dictionary i Python
- Lagrer pekere til objekter uten en bestemt rekkefølge (et sett).
- Består av nøkkel-verdi-par
- Hvert objekt som legges inn må ha en tilhørende nøkkel for å gjøre oppslag i hashmapen.
- Noen metoder:
  - **put**(key, value)
  - **get**(key)
  - **remove**(key)
  - **keySet**()

```
Katt k1 = new Katt(navn:"Ole");
Katt k2 = new Katt(navn:"Dole");
Katt k3 = new Katt(navn:"Doffen");

HashMap<String, Katt> kattene = new HashMap<>();
kattene.put(k1.navn, k1);
kattene.put(k2.navn, k2);
kattene.put(k3.navn, k3);
```

For each over HashMap

```java
Katt k1 = new Katt(navn:"Ole");
Katt k2 = new Katt(navn:"Dole");
Katt k3 = new Katt(navn:"Doffen");

HashMap<String, Katt> kattene = new HashMap<>();
kattene.put(k1.navn, k1);
kattene.put(k2.navn, k2);
kattene.put(k3.navn, k3);

//for each på hashmap - itererer over settet med nøkler
//for å aksesere verdien knyttet til nøkkelen
for (String kattenavn : kattene.keySet()) {
    System.out.println(kattene.get(kattenavn));
}
```

```
Katt@6b95977
Katt@7e9e5f8a
Katt@8bcc55f
```

For each over HashMap

```java
Katt k1 = new Katt(navn:"Ole");
Katt k2 = new Katt(navn:"Dole");
Katt k3 = new Katt(navn:"Doffen");

HashMap<String, Katt> kattene = new HashMap<>();
kattene.put(k1.navn, k1);
kattene.put(k2.navn, k2);
kattene.put(k3.navn, k3);

//for each på hashmap - itererer over settet med nøkler
//for å aksesere verdien knyttet til nøkkelen
for (String kattenavn : kattene.keySet()) {
    kattene.get(kattenavn).spis();
}
```

```
jeg spiser kattemat
jeg spiser kattemat
jeg spiser kattemat
```

Oppgave 1a: klasser og objekter

- Følgende klasseskjelett er gitt:

```
class Kaningaard {

    private Kanin [] kaniner = new Kanin[100];

    public boolean full() { . . . }
    public boolean tom () { . . . }
    public Kanin finnEn(String navn) { . . . }
    public void settInn (Kanin kanin) { . . . }
    public void fjern(String navn) { . . . }
}
```

- Lag en klasse Kanin. Kaninen har et navn som angis ved opprettelse. I tillegg har klassen en metode, hentNavn(), som returnerer navnet til kaninen.

Oppgave 1b

- **Skriv ferdig klassen Kaningaard ved å fylle ut metodene i klasseskjelettet gitt ovenfor. Du kan gå ut fra at alle kaniner skal ha forskjellige navn.**
- Ta derfor høyde for at det ikke er lov å sette inn en kanin to ganger (altså med samme navn som en i arrayen).
- Ta også høyde for noen andre problemer som kan oppstå og gi gode feilmeldinger.
- To eksempler på problemer er dersom man forsøker å sette inn en kanin i en full kaningård, eller når man forsøker å fjerne en kanin som ikke er i listen.
- Du kan anta at klassen Kanin inneholder metoden hentNavn().

Oppgave 1c

- Skriv et hovedprogram der du gjør minst tre tester av de forskjellige delene av klassen Kaningaard. Test for eksempel tilfeller som innsetting av kanin ved full kaningård eller fjerning av en ikke-eksisterende kanin.

Oppgave 3: Array, ArrayList og løkker

a) Lag et program med en array med plass til 10 heltall. Fyll arrayen med alle sifrene mellom 0 og 10 i stigende rekkefølge. Skriv ut alle verdiene på hver sin linje.

b) Endre programmet slik at du får utskriften i motsatt rekkefølge. Denne gangen skal du bruke en ArrayList i stedet for en array.

c) Endre programmet fra a slik at utskriften nå er tallene i arrayen eller ArrayList-en og summen av alle tallene som er skrevet ut til nå. Denne gangen skal du bruke en "for each"-løkke.

d) Endre programmet slik at tallet blir gjort om til en streng og at tallene blir lagt til i en string som samler alle tallene per iterasjon. Her velger du om du vil ta utgangspunkt i arrayen eller ArrayList-en, og du velger mellom vanlig for-løkke og "for each"-løkke.

e) Lag et program som oppretter en array med plass til 10 strenger. Fyll arrayen med *, hvor plass/indeks 0 har én * og hvor plass/indeks 9 har ti *-er. Det vil si at hver plass inneholder (plassnummer + 1) antall *-er. Skriv ut alle strengene på separate linjer.