Forelesning 6

INF101 - 31/1 2022

UNIVERSITETET I BERGEN

Hva har vi lært om Objekter

- En Klasse definerer en type objekter.
  - Definert av metoder og variabler
- Objekter lages fra en klasse
  - «new» brukes for å lage nye Objekter fra en klasse
  - Konstruktør beskrive hvordan Objekter lages
- Eksempel, en klasse «Book», en bok har
  - Tittel, forfatter, årstall og antall sider
  - Hvert bok objekt har forskjellige verdier

Hva har vi lært om Objekter

- Variabler er referanser/pekere til objekter
  - Flere pekere kan peke på samme Objekt
  - Variabler av primitive typer er ikke pekere
- Statiske metoder bruker ikke variablene i et objekt, men tilhører klassen

Objekt terminologi

- Type
  - hvilken klasse ble brukt til å lage objektet
- Behavior
  - Hvilke metoder kan brukes på Objektet og hva gjør disse metodene?
- Identity
  - Hvilket objekt er det? (Minneadresse)
- State
  - Hvilke data lagrer Objektet akkurat nå

Eksempel

- La oss implementere en enkel klasse Book

Objektorientert programmering

Abstraction

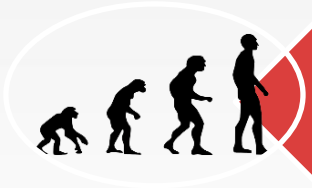
- Fokuser på det viktigste og gjør det enkelt

Modularity

- Lag kode som er byggeklosser

Encapsulation

- Skjul detaljer, koden blir lettere å bruke/tryggere



Arv/polymorphism

- Arv - Gjenbruk av klasser
- Polymorphism – metoder virker på flere typer

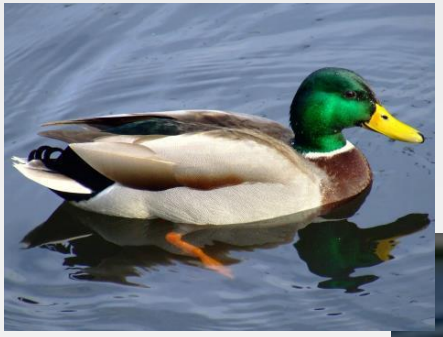
Abstraksjon

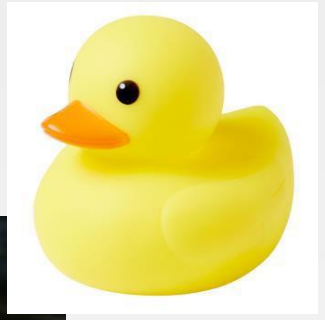
- Når man skal lage et program er det ofte begreper og objekter i virkeligheten som skal representeres som objekter i koden.
- Vi representerer ikke alle detaljer, bare det viktigste.

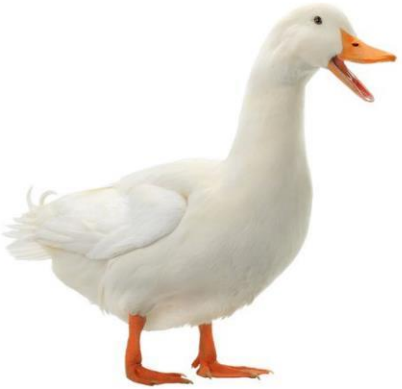
Abstraksjon

- En hver klasse har et visst ansvar, å legge rett funksjonalitet til rett klasse er viktig.
- Når rett funksjonalitet er i rett klasse blir systemet lett å bruke og lett å forstå
- Ikke for mye detaljer, ikke for lite funksjonalitet.

Abstraksjon







Sitat – duck-typing

James Whitcomb Riley (1849–1916):

“When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.”

Duck-typing betyr at vi trenger ikke alltid beskrive virkelige objekter nøyaktig, bare de egenskapene som er nødvendige.

Modularity

- Modularitet betyr å dele koden inn i deler som lett kan byttes ut.
  - Klasser, funksjoner, pakker osv.
  - I morgen skal vi lære om Interface som er en viktig måte å oppnå modularitet
- Real-world eksempler:
  - Et USB tastatur fungerer uansett om du har PC eller macbook.
  - Et dekk kan brukes på mange forskjellige biler
  - En myntenhet fungerer i mange banker

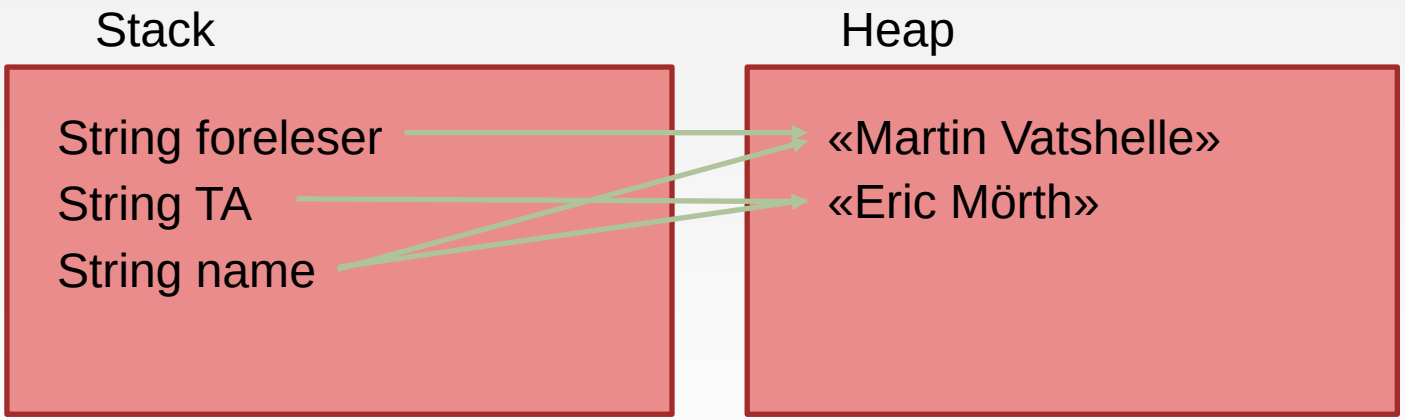
Objekt klassen

- Alle klasser er av typen Objekt
  - Vi lærer mer om dette når vi kommer til arv
- Noen viktige metoder finnes i alle klasser
  - toString()
  - equals()

- Forrige uke lærte dere forskjellen på equals() og ==

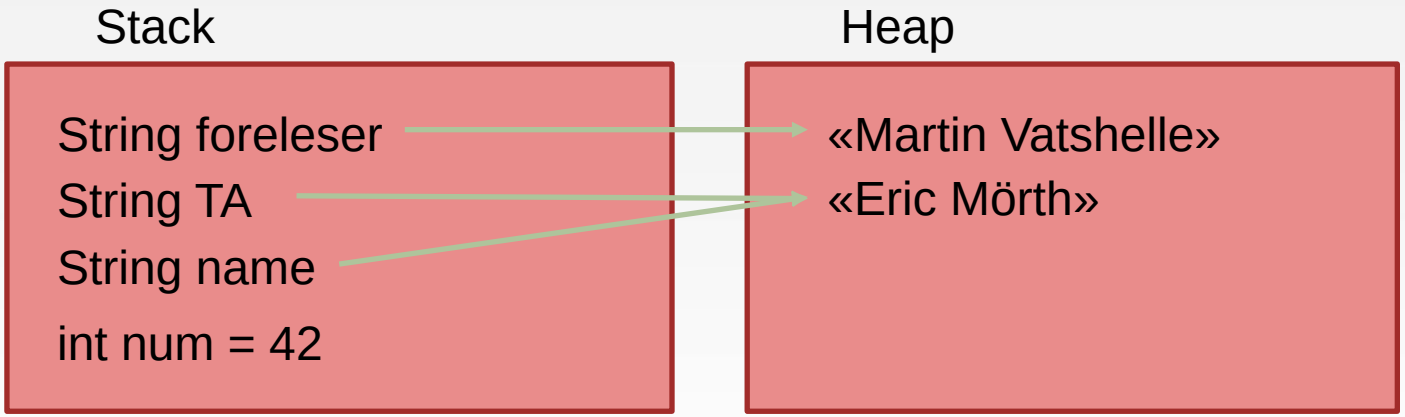
Java minne

- Java bruker ikke hele RAM
  - Stack lagrer variabler/referanser (~1 MB)
  - Heap lagrer data/objekter (~1GB)



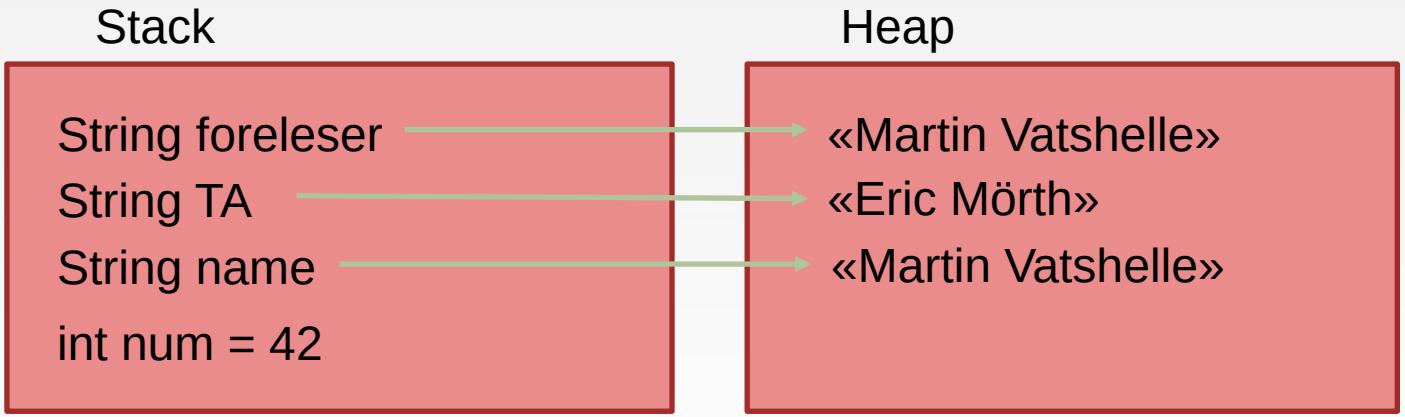
Primitive variables

- Lagres på Stack, ikke Heap fordi de tar liten plass
- int, boolean, float, double, char…
  (alle typer som starter med liten bokstav)



Primitive variables

- Lagres på Stack, ikke Heap fordi de tar liten plass
- int, boolean, float, double, char…
  (alle typer som starter med liten bokstav)



Java Garbage Collection

- Noe data på heapen er i bruk.
- Noe er vi ferdig med.
- Java finner automatisk ut hvilke data som kan slettes fra Heap

La oss implementere Bibliotek

- Biblioteket trenger et system for å holde styr på alle bøkene sine.
- De har hørt at vi har laget en Book klasse
- Kan vi hjelpe dem?

== operator i Java

- Sjekker kun på Stack
- Objekter er sammenlignet på om de peker til samme minneadresse på Heap
- To objekter kan ha helt likt innhold men være lagret på forskjellige minneadresser

equals() method

- Equals metoden er implementert i Object.java

```
public boolean equals(Object obj) {
    return (this == obj);
}
```

- Ofte ønsker man å overskrive equals() metoden slik at den sjekker at innholdet er likt.

- La oss prøve ut litt av dette på Book

Java lagrer andre steder også

- String pool – String er spesiell og JVM gjør noen optimiseringer for String
- Integer objekter med verdier mellom -128 og +127 er også lagret spesielt
- JVM heap – definisjoner av objekt typer og deres metoder

```
public class Thanks{
    private String text;

    public Thanks(String text){
        this.text = text;
    }

    public String toString(){
        return this.text;
    }

    public static void main(String[] args){
        Thanks end = new Thanks("Takk for forelesning!");
        System.out.println(end);
    }
}
```