Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

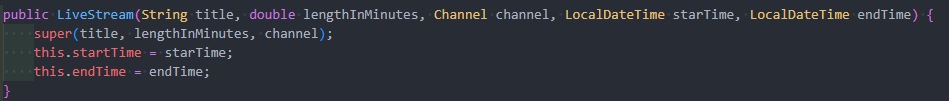
Automatisk generert beskrivelse**Definere noe:**

* **Static**: Brukes kun i klassen du lager variabelen.
* **FINAL**: Variabel for en fast verdi, eks. PI (3.14). Hvis en klasse er final, kan den IKKE arves. Kan ikke endres på i ettertid, hvis den er initialisert.
* **ABSTRACT:** Metoder som ikke har en implementasjon. Disse må arves av andre klasser og defineres ved å først bruke extends på klassen.
* **Void**: Ingen return-statement
* **Public**: Gjør innhold tilgjengelig for alle klassene i Java-prosjektet
* **Private**: Gjør innhold kun tilgjengelig i klassen det er definert i
* **Protected**: Gjør noe tilgjengelig for klasser som arver fra *denne* klassen (klassen som oppretter variabelen f.eks.)

**Struktur, sammenkoblinger og oppsett av Java:**

* **Interface**: Definer hva som skal gjøres, men ikke hvordan. Inkluder eventuelle parametere. (<T> viser til at man kan velge listen selv = Generic Interface). EN klasse kan implementerer FLERE interfaces. Den implementerende klassen må skrive «implementens <interfaceNavn>» og deretter implementerer metodene med @override
* public interface ObjectFileHandler<T>{
* public void writeObjectsToFile(String filename, ArrayList<T> objectsToWriteToFile);
* public ArrayList<T> readObjectsFromFile(String filename);
* }
* **Models**: Klasser som har konstruktører, metoder og objekter. Metodene kan kalles på/brukes i main. Starter med: Public class <klasseNavn> {
* **Tools**: Verktøy med definert handling og struktur. F.eks. writeObjectsToFile(fileName) eller readObjectsFromFile(fileName). Her vil filnavnet være parameter. Enten så skriver man til filen, eller så leser man fra den.

**Begreper for ulike ting i et Javaprogram:**

* **Objekter:** Instanse av klasse. (Klasse, variabelnavn blir til objekt)
* **Konstuktører (m. super):**
* **Metoder:** Eks. «public double getSaldo». Likner på funksjoner i Python. Gjør en handling, med eller uten parametere.

Et bilde som inneholder tekst, Font, skjermbilde

Automatisk generert beskrivelse**Kodestruktur -og logikk:**

* **MAIN** (kun i filen hvor man skal kjøre kode. «Main.java»):
  + Ikke implementer nye metoder her. I såfall må disse være static
* **For-each:** klasse, individuelle elementer : liste som skal gjennomgås
  + for (Star starX : starsFromFile)
* **printf:** Bestem antall desimaler: (% brukes for det som skal bestemmes - variabel)
  + System.out.printf("På månen ville min vekt vært %.2f kg" + "\n", minMåneVekt);
* **Array:** Opprett en «liste» med et gitt antall indexer, som man senere kan tildele (må være samme datatype:
* **Et bilde som inneholder tekst, Font, skjermbilde

  Automatisk generert beskrivelse**To av flere måter for å opprette array 🡪
* **ToString()/Override:** Brukes når man vil standardisere en utskrift. Kalles på slik (solarSystem er et objekt av en klasse):
* Et bilde som inneholder tekst, Font, skjermbilde, Grafikk

  Automatisk generert beskrivelseSystem.out.println(solarSystem);

****

* **ArrayList:**
  + **Legg til objekt i lista:**
  + **Antall elementer i lista:**
  + **Hent ut:** «.get(<index>)
* **Extends/Arv:** Arver fra en overklasse. Husk super i konstruktør.
  + Mindre redundant kode
  + Greit å bruke for å oppdatere og vedlikeholde, grunnet felles egenskaper i ulike klasser.
* **Implements:** Brukes når man skal implementere noe fra et interface

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

* **Typecasting:** Konvertere verdier. **🡪**
* **Instanceof:** Sjekker om en klasse tilhørerer et gitt objekt.

Et bilde som inneholder tekst, skjermbilde, Font, line

Automatisk generert beskrivelse

**Exceptions**:

* **Checked Exceptions**: Forventede feil som oppstår – feil som vi MÅ ta stilling til
* **Runtime Exceptions**: Feil som oppstår under kjøring, som er vanskelig å forvente (logikkfeil)
* **Errors**: Alvorlige feil – typisk ikke håndterbare.
* Et bilde som inneholder tekst, skjermbilde, Font

  Automatisk generert beskrivelse2 siste = Unchecked exceptions.
* **Try & catch:**
* **Finally:** Kodeblokk som alltid vil bli kjørt uavhengig om koden kjører som den skal, eller ikke.

Et bilde som inneholder tekst, skjermbilde

Automatisk generert beskrivelse**Filbehandling**:

**Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelseEt bilde som inneholder tekst, skjermbilde, programvare, display

Automatisk generert beskrivelse**Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

Automatisk generert beskrivelse

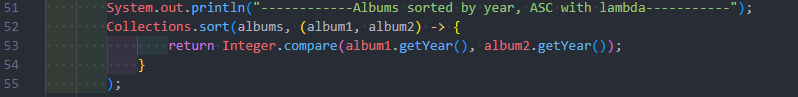
**Filspråk:**

* Et bilde som inneholder tekst, skjermbilde, Font

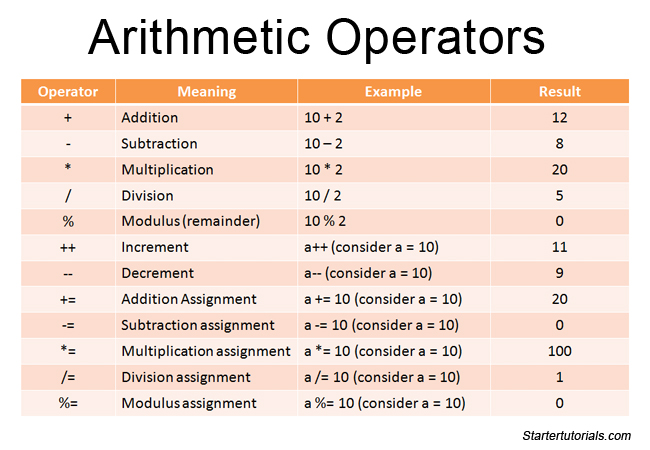
  Automatisk generert beskrivelse**XML**: Exstensible Markup Language 🡪
  + Likner på HTML tagger
  + Benytter start og sluttagger:
* **JSON**: JavaScript Object Notation
  + Hierarisk strukturert
  + Benytter: «key: value» struktur
  + Et bilde som inneholder tekst, Font, skjermbilde, line

    Automatisk generert beskrivelseEksempel: 🡪

**Lambda vs uten Lambda:**



**Et bilde som inneholder tekst, skjermbilde, nummer, Font

Automatisk generert beskrivelseEt bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse**