

# Oblig 1 – Programmering 2

Thomas Waaler

## Oppgave 1.1

JRE, altså Java Runtime Environment, er hva som trengs for å kjøre Java applikasjoner. JRE er det eneste sluttbrukeren trenger. Siden Java blir kjørt i en VM, ved bruk av Java Virtual Machine, er det enklere å utvikle for flere type plattformer.

Dersom man skal utvikle Java applikasjoner trenger man JDK, Java Development Kit, som inneholder alle verktøy man trenger for utvikling. Denne pakken inneholder også JRE, så da trenger man bare å skaffe seg JDK.

## Oppgave 1.2

Flyten av hvordan Java applikasjoner blir bygget og kjørt er som følgende; Si vi har en Java source fil kalt MyFirstProgram.java. For å kunne kjørt dette programmet må vi først kompilere, som vi kan gjøre ved å skrive følgende i terminalen:

```
javac MyFirstProgram.java
```

Dette vil gi oss en bytekode fil kalt MyFirstProgram.class, som er leselig for JVM'en. Deretter for å kjøre programmet vårt kan vi skrive følgende i terminalen:

```
java MyFirstProgram
```

## Oppgave 1.3

Compile-time error er feil som oppstår når man kompilerer programmet. Oppstår når man har feil i syntaks og/eller semantikk. For eksempel:

```
string myString = "This is my string!";
```

Her vil vi få error når vi prøver å kompilere siden det skal være String og ikke string.

En Runtime error er feil som kan oppstå selv etter at programmet er kompilert riktig. Dette er vanligvis logiske feil som ikke har blitt plukket opp under kompilering eller noen andre ting som at man forsøker å dele på 0 osv. Et eksempel på runtime error kan være:

```
for (int i = 5; i > -5; i--) {  
    int newNum = 100 / i;  
    System.out.println("100 devided by " + i + " = " + newNum);  
}
```

Her vil vi bare få en error når vi prøver å kjøre koden og variabelen i har inkrementert til 0. Da vil man få en ArithmeticException siden man prøver å dele 100 på 0, som ikke er mulig.

## Oppgave 1.4

En klasse er en mal for hvordan et objekt skal være satt opp og hvordan den virker. Eksempel på en klasse som har ett privat variabel som har getter og setter, og klassen har en konstruktør som instanserer variabelen.

```
class MyClass
{
    private String someText;

    public MyClass() {
        someText = "Placeholder text";
    }

    public void setSomeText(String text) {
        someText = text;
    }

    public String getSomeText() {
        return someText;
    }
}
```

Men å ha en klasse gjør ingenting med mindre vi bruker den klassen til å opprette ett objekt. Først lager vi et variabel som bruker MyClass som datatypen og instanserer konstruktøren ved bruk av `new`. Da har man opprettet et objekt og kan kalle metodene osv.

```
MyClass myClass = new MyClass();
myClass.setSomeText("Some other text!");
System.out.println( myClass.getSomeText() );
```