

Teoridel – oblig1

Oppgave 1.1

- JRE er en programvareplattform som brukes for å kjøre Java-applikasjoner. Denne plattformen inneholder også JVM (Java Virtual Machine). JRE er altså et miljø for å kjøre kode som allerede er kompilert (.class filer).
- JDK (som vi bruker i IntelliJ IDEA) er utviklingssettet som brukes for å skrive, kompilere og kjøre Java-kode. Denne inneholder både JRE og JDK, som trengs når man skal både skrive og kjøre programmer.

Oppgave 1.2

- Flyten av hvordan Java-applikasjoner blir bygget og kjørt, er slik at koden først blir skrevet i valgt utviklingsmiljø (f.eks IntelliJ IDEA), så blir kildekoden oversatt til bytekode (kompilert) ved hjelp av javac (kompilatoren). Denne genererte bytekoden lagres som en .class fil kan kjøres ved hjelp av java (Java Virtual Machine) som tolker og kjører denne.
- TL:DR: Kildekode med .java -> Kompilering med javac -> Bytekode .class -> Kjøring med java (JVM).

Oppgave 1.3

- Compile-time errors oppstår under kompilering, altså før programmet blir kjørt. Dette forekommer ved feil med syntaks eller at andre Java-regler ikke er tilfredsstilt.
- Run-time errors oppstår etter programmet er kompilert og kjører, og kan skje selv om koden er kompilert uten feil. Dette forekommer som oftest ved logiske feil, ugyldige inputs eller for eksempel ved deling på null.

Oppgave 1.4

- Klasse: En klasse kan sees på som en «blueprint» eller en oppskrift. Oppskriften inneholder ulike datatyper som variabler (instansvariabler) for å opprette et objekt basert på disse.
- Objekt: Et objekt kan sees på en instans av en klasse. Med klassen som er forklart over, kan man bruke disse instansvariablene for å opprette helt unike objekter med disse. Man kan bruke klassen til å lage uendelig mange objekter.
- Eksempel: I en klasse «Student», kan to instansvariabler f.eks være fornavn og etternavn. Objekter kan da opprettes ved bruk av Student klassen til å lage unike objekter med unike fornavn og etternavn.