Class

Class er en måte og lage en mal(blueprint) på et objekt. Trenger ikke å lage ny kode for hvert objekt, Men man kan lage et nytt objekt basert på blueprinten.

Object (konseptet, ikke klassen)

Konseptet med objekter er å få mer dynamisk kode. Når man skal lage noe flere ganger, kan man gjøre det til et objekt. Når noe er et objekt, kan man lage en "datatype" slik at man kan lagre mer "dynamisk" data.

Dette betyr også at man ikke trenger og skrive mye kode, fordi du allerede har mange funsjoner i klassen.

Instansvariabel

Dette er variablene man ggir til et objekt, de er definert i klassen.

Siden man definerer det i klassen, vil alle objekter fra samme klasse ha samme instansvariabler som man kan nå når du leser objektet.

Overloading

Dette er når du har en metode, men du definerer flere metoder med forskjellig antall argumenter.

Dette er nyttig hvis du vil generere et objekt, Men du vet ikke om du har all informasjonen man trenger når man sakal generere objektet.

Overriding

Overrriding er når man lager en funksjon i en lavere klasse (en klasse bygget på en annen klasse) of funksjonen allerede eksisterer.

Når du lager en ny funksjon, vil JAVA erstatte den gamle funksjonen med den nye funksjonen , så man har overskrevet den andre variabelen/metoden.

Extends

Hvis du skal lage flere typer objekter som skal ha noe felles kan man lage en generell klasse, deretter lag andre klasser og be dem "extend" den generelle klassen.

Når du gjør dette, får klassen som utvider den generelle klassen få sammen informasjon (funksjoner, instansvariabler osv.), Man kan også lage spesefike funksjoner og instansvariabler som bare denne klassen kan bruke.

Polymorphism

Dette er en måte å få objekter til og utgi seg for å være fra klasser over.

Dette lar oss kombinere lavere klasse med større klasse.

Om man vil sammenligen en planet med solen, må man gjøre planeten og solen over til en felles klasse som de arver fra, da blir de til samme objekt (og samme "datatype").

private,public,(protected) (klasse,variabel,metode)

- Private bruker man når man ikke vil ha noe "synlig" for andre uten for klassen. Hvis man bruker en privat variabel kan den ikke bli sett/endret fra en annen funksjon uten for klassen.

- Public: Lagrer man noe som public kan man se/endre på variabelen utenfor klassen. Det betyr at når man har en variabel som man vil at en funskjon skla kunne endres på i en klasse vil man kunne endre den.

-Protected Dette lar deg endre variabler fra andre steder under klassen, men ikke fra over. Hvis du hae wn funksjon i main som man vil endre på en variabel, kan man ikke kunne gjøre det med mindre man endrer til public.

Disse er relaterte med tilgangs nivåer.

Når man har noe public kan man nå det fra alle steder.

Når man har noe protectet kan man nå det fra klasser, packer og underklasser, men utenom fra main.

Hvis noe er private kan bare klassen nå den.

this og super

This: er nåe du vil referere til noe i samme klasse, så hvis du har en funksjon med et argument, kan argumentet ha samme navn som en variabel i samme klasse, da når variabelem i klassen og ikke argumenter kan man sette this. Forran

Super referer til noe som ligger i overklassen. Om klassen arver noe fra en klasse kan man bruke super() for og referere til noe som er i klassen over.