Tors 17. feb 2022

Private / Protected / Public (Metoder og variabler)

Bestemmer om variablene og metodene skal bare være tilgjengelige i klassen (private), for klassen og subklassene (protected) eller også utenfor klassen (public).
Forskjellige tilgangsnivåer. Private kan kun brukes innen den gjeldende klassen, protected kan i tillegg brukes innen den gjeldende klassens subklasser, og public kan brukes av alle.
Private - Variabler eller metoder som ikke skal kunne refereres til eller brukes utenfor klassen. Protected - Variabler eller metoder som kun skal refereres til av klassen og dens subklasser. Public - Metoder tilgjengelig i klassens grensesnitt.

Static vs. non-static (Metoder og variabler)

Static variabel - Har den samme verdien på tvers av alle objektene av en klasse. Static metode - Metoder som kan brukes uten å opprette et objekt av klassen.
Static metoder og variabler vil være likt for alle objektene av den klassen, også dersom de endres. Dvs at en static variabel med verdi int 1 vil ha samme verdi i alle objektene som bruker den. Om et objekt endrer på verdien endres den i alle.
Metoder: Brukes for å kunne kjøre metoder som ikke hører til et faktisk opprettet objekt. Variabler: Static er konstante(variabel) for en klasse

Abstract (Klasser og metoder)

Abstract klasser - klasser du ikke kan opprette objekter av. Man kan kun arve egenskapene i andre klasser. Abstract metode - En metode som skal brukes av alle subklassene, men den har ikke noe kodeblokk/implementering. Abstrakte metoder er må defineres i underklasser.

Klasser (Superklasse, Subklasse, Super-nøkkelordet)

Subklasser arver egenskaper fra superklasser Super er et nøkkelord som gir tilgang til superklassens metoder, variabler eller konstrøktør

Type på pekere

- Hvilke objekter kan en peker av en gitt type inneholde?
- Hva bruker vi instanceof til?
- Hva er casting? Og hvorfor er det nyttig?

En peker kan inneholde objekter av den typen/klassen pekeren er, pluss alle dens subklasser instanceof sjekker om et objekt har egenskaper (dvs. har arvet) fra en gitt klasse casting endrer typen til en peker

Tors 17. feb 2022

Override (overskriving)

Endrer implementasjonen til en tidligere definert metode
Overskriver en arvet metode-definisjon
Override (norsk: overskriving) skriver over metoden som blir arvet fra superklassen og implementerer noe ekstra/annerledes i denne metoden, som enten er arvet fra en superklasse eller et interface

Overloading (overlasting)

Deklarere en metode flere ganger, med forskjellig antall/type parametere (signatur) og eventuelt forskjellig kodekropp
Overloading - Når man har flere metoder i en klasse med samme signatur, men forskjellige parametere.
Metoder med samme navn med ulike argumenter

Interface

Slags huskeliste der flere klasser, både super- og subklasser kan arve fra, hovedsakelig metoder, som skal bli overskrevet, i klassene som arver fra interfacet.
En samling metodedefinisjoner som også kan «arves»

Abstract class vs. interface

Hva er forskjellen mellom abstract class og interface? Når må vi bruke interface?

En abstract class kan ha variabler, og implementere metoder som en vanlig klasse, dette kan ikke interface. Abstract class kan også ha abstract metoder, som ligner litt på metode-deklarasjonene man finner i interface (klasser som extends/implements må implementere disse metodene) Interface brukes for å samle lik oppførsel. En klasse kan arve fra mange interface/grensesnitt, mens den kan kun arve fra én klasse.

Extends vs. implements

Når bruker vi hver av dem?

“Extends” brukes når man arver fra klasser, “implements” brukes man arver interface/grensesnitt
“Implements” er mellom “interface” og “klasse”, “extends” er mellom to av samme type.
class extends class class implements interface interface extends interface