

Trinn 7 – Static, final og arv

Det er vel på tide å komme inn på et annet hovedemne innen objektorientert programmering også: Arv!

Også denne gangen har jeg laget en lengre intro-video, så se gjerne den før du begynner.

Mål for dette trinnet:

- Jeg kan lage en klasse som arver fra en annen klasse.
- Jeg vet hvordan jeg kan override en metode.
- Jeg forstår hva static innebærer for fields og metoder.
- Jeg forstår hva final innebærer.
- Jeg kjenner til Object-klassen og noen av metodene som ligger der.

Relevante kapitler i pensumboka:

- Kap 10: Static methods and variables
- Kap 14: Overriding methods

[Her](#) er spørreskjemaet for trinn 7 der du kan fortelle hvordan det gikk.

Oppgave 1

Du skal lage 4 klasser. Du må selv finne ut hvordan klassene forholder seg til hverandre når det gjelder arv. Her er klassene:

Shape (geometrisk figur):

- Fields:
 - o *color* - Hvilken farge som er satt på figuren (shape). Datatype: java.awt.Color.
 - o *filled* - Om figuren er fargelagt eller ikke.
- To konstruktører
 - o Tom (ingen parametre). Denne skal sette figuren til fargelagt med rødfarge.
 - o En som tar både *color* og *filled* og setter fields med verdiene.
- Gettere og settere
- En *toString*-metode som returnerer en tekstlig beskrivelse av objektet.

Circle (sirkel):

- Field: *radius*. La oss bruke den primitive data-typen *double*, så får vi erfaring med den.
- Tre konstruktører:
 - o Tom. Setter radius til 1.0. Arver standardverdier fra *Shape*.
 - o En som tar *radius*. Arver standardverdier fra *Shape*.
 - o En som tar *color*, *filled* og *radius*.
- En *toString*-metode som returnerer en tekstlig beskrivelse av objektet.
- En *getArea*-metode som returnerer arealet til figuren.
- En *getPerimeter*-metode som returnerer omkretsen til objektet.
- Getter og setter

Rectangle (rektangel):

- Fields:
 - o *width* – bredden til rektangelet (double)

- *length* – lengden til rektangelet (double)
- 3 konstruktører
 - Tom. Setter bredde til 1.0 og lengde til 2.0 og fylt med grønn farge.
 - En som tar *width* og *length*. Setter fylt med grønn farge.
 - En som tar *width*, *length*, *color* og *filled*
- Gettere og settere
- En *getArea*-metode som returnerer arealet til figuren.
- En *getPerimeter*-metode som returnerer omkretsen til objektet.
- En *toString*-metode som returnerer en tekstlig beskrivelse av objektet.

Square (kvadrat – en type rektangel der alle sider er like lange)

- Field: *side* – lenden på sidene i kvadratet (de er jo alle like lange) (double)
- 3 konstruktører
 - Tom. Setter side til 1.0.
 - En som tar *side*
 - En som tar *side*, *color* og *filled*

La oss si at ettersom *Square* er en type *Rectangle*, så skal også *Square* ha standardverdier «fylt med grønnfarge» hvis verdiene ikke presiseres ved opprettelse av objektet.

- Getter og setter
- En *getArea*-metode som returnerer arealet til figuren (hvis du trenger det).
- En *getPerimeter*-metode som returnerer omkretsen til objektet (hvis du trenger det).
- En *toString*-metode som returnerer en tekstlig beskrivelse av objektet.

Oppgave 2

Lag et program som tester om klassene du har laget fungerer tilfredsstillende. Sørg for å opprette de ulike objektene med forskjellige konstruktører og kall på metodene som *toString*, *getArea* og *getPerimeter* for å se om du får forventet resultat.

En del av testingen innebærer å sjekke om riktige standardverdier er satt når de benyttes ved opprettelse av et objekt. Her er et lite sammendrag:

- Standardverdier for *color* og *filled* for geometriske figurer er rød og at den er fylt.
- Standardverdi for radiusen til en sirkel er 1.0.
- Standardverdier for rektangler er fylt med grønn og lengde=2.0 og bredde=1.0.
- Standardverdi for sidene i et kvadrat er 1.0.

Oppgave 3

Ser det ut som om klassene du har laget fungerer tilfredsstillende? Så bra! Men hvorfor ikke sjekke om testene til Per viser det samme? Bruk test-klassene som ligger i Canvas og se om klassene dine fungerer slik testene forventer. Gjør eventuelle nødvendige utbedringer.

Du bruker test-klassene på samme måte som vi gjorde i Trinn 2, så ta en titt der hvis du trenger informasjon om bruk av test-klassene.

Testene forutsetter at getter-metodene dine har visse navn (*isFilled*, *getLength*, *getArea* osv.). Disse er basert på IntelliJ's autogenerering av gettere og settere. Så hvis du har benyttet autogenerering, så vil det gå bra (så lenge du har samme navn på fields som beskrevet i oppgaven).

Ekstraoppgave 1

Lag en ny klasse for en valgfri geometrisk figur. Lag dine egne regler for standard-verdier.

Ekstraoppgave 2

Se på de automatiserte testene fra Canvas. Forsøk å lage tilsvarende tester for din nye klasse i Ekstraoppgave 1. Dette har vi ikke gått igjennom, men jeg tror det skal være tilstrekkelig informasjon å støtte seg på i de eksisterende testene.

Ekstraoppgave 3

Har du enda mer tid til overs? Jøss, så bra! Vi har snakket om muligheter for å finne flere oppgaver på nett. Mengdetrening er gull hvis man vil bli god til å programmere. Jeg vet at flere av dere allerede bruker [code wars](#). Hvorfor ikke prøve ut den?

Et annet tips jeg har fått fra en student er [Kattis](#). Nå som vi har lært å lese fra fil er det mange oppgaver der som dere har mulighet til å løse. Som for eksempel:

- <https://open.kattis.com/problems/everywhere>
- <https://open.kattis.com/problems/aaah>
- <https://open.kattis.com/problems/abc>

Oppgavene har en vanskelighetsgrad, så du kan finne oppgaver som du mener passer for deg. Finner du en fin oppgave? Del den gjerne i Discord!