

## Laboration 10 – TurtleRace, del 2: mer arv

*Mål:* Du ska lära dig mer om att skriva och använda klasser som är strukturerade med hjälp av arv. Du får använda polymorfism – arv – för att beskriva olika slags sköldpaddor.

### Förberedelseuppgifter

- Gör hela TurtleRace del 1. Det är viktigt att den fungerar innan du sätter igång.
- Läs i läroboken: Downey & Mayfield, kapitel 14.
- Läs avsnittet Bakgrund.

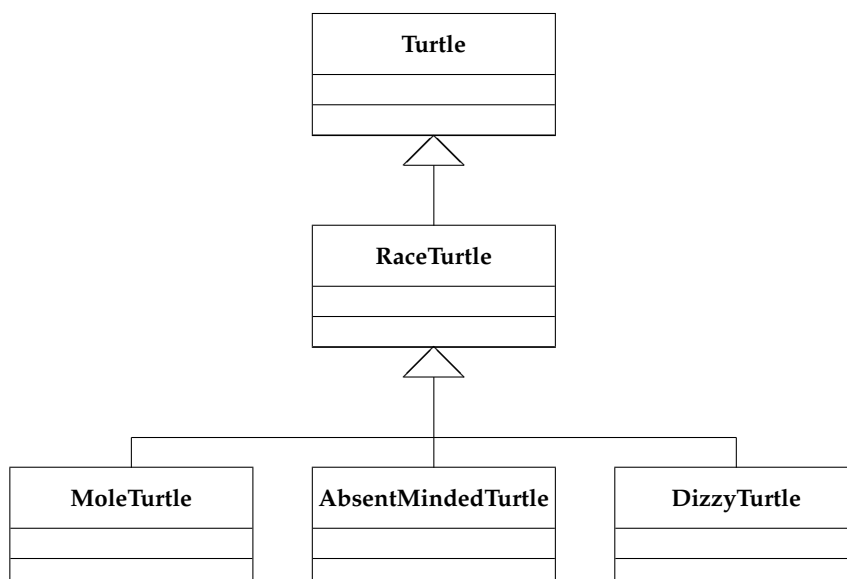
### Bakgrund

Under den här laborationen ska du skriva ett program där du låter olika typer av sköldpaddor tävla mot varandra i en kapplöpning. Denna laboration bygger vidare på din laboration 9.

Uppgiften går ut på att skriva ett antal klasser som beskriver olika slags tävlande sköldpaddor, samt att modifiera huvudprogrammet för att genomföra en sådan kapplöpning. Klassen `RaceTurtle` (från laboration 9) ska representera det som samtliga tävlingssköldpaddor har gemensamt. Nu ska emellertid metoden `raceStep` överskuggas (override) av subclassernas mer specifika definition av metoden. Vi behöver tre subclasser, en för varje slags sköldpadda, och bilden nedan visar arvshierarkin.

Programmet ska innehålla tre slags tävlande sköldpaddor (subclasser till `RaceTurtle`):

- `MoleTurtle`, som beskriver en "mullvadssköldpadda", d.v.s. en tävlande sköldpadda som då och då går under jorden (genom att lyfta pennan slumpmässigt).
- `AbsentMindedTurtle`, som är en tankspridd sköldpadda. Graden av tankspriddhet anges i procent när sköldpaddan skapas, och sannolikheten att en tankspridd sköldpadda skall gå framåt bestäms av tankspriddhetsgraden. Exempel: en tankspriddhetsgrad på 34 procent ska medföra att sköldpaddan i 34 procent av fallen glömmer att ta sitt steg.
- `DizzyTurtle`, som beskriver en yr tävlingssköldpadda, som vinglar när den skall ta sig framåt. När dessa sköldpaddor skapas ska graden av yrsel (från 1 till 5) anges, och deras förmåga att hålla kursen skall bestämmas av yrselgraden. Du behöver inte göra någon avancerad simulering av yrseln, men en sköldpadda med högre yrselgrad bör vingla mer än en mindre yr sköldpadda.



## Datorarbete

1. Kopiera filerna från förra laborationen genom att högerklicka på projektet *Lab09* och välja *Copy*. Högerklicka sedan igen och välj *Paste*. Ge projektet ett nytt namn, t.ex. *Lab10*. Öppna det nya projektet och fortsätt din utveckling där.
2. Implementera tävlingssköldpaddorna som beskrivs ovan. Skapa en ny klass för varje typ av sköldpadda och låt varje klass ärva från *RaceTurtle*. Varje sådan subclass ska ha sin egen definition av metoderna *raceStep()* och *toString()* – dessa ska alltså implementeras på nytt.

**Observera** att *raceStep()* och *toString()* inte är abstrakta i *RaceTurtle*. De har alltså redan en implementation. Det gör att du själv måste komma ihåg att överskugga dessa i varje subclass, men de går samtidigt att utnyttja för att slippa duplicera kod. Exempelvis kan man från subclassernas *raceStep*-metoder anropa *super.raceStep()* för att genomföra själva steget, förutom den kod som är specifik för varje subclass (lyfta/sänka pennan, jämföra tankspriddhet, välja riktning).

3. Modifiera klassen *TurtleRace* för att genomföra ett lopp enligt följande:
  - Skapa åtta sköldpaddor av slumpmässig typ.
  - **Notera** att då alla ärver från en gemensam klass behöver listan inte ändras på något vis. Se till att du förstår varför en lista av *RaceTurtle* även kan innehålla objekt av klasserna *MoleTurtle*, *AbsentMindedTurtle* och *DizzyTurtle*. Fråga gärna handledaren.
  - Då *AbsentMindedTurtle* ska skapas ska tankspriddhetsgraden slumpas fram mellan 0 och 100.
  - Då *DizzyTurtle* ska skapas ska yrselgraden slumpas fram mellan 1 och 5.
  - Innan loppet börjar ska hela startuppställningen skrivas ut enligt exemplet nedan, ett krav är att detta sker genom respektive sköldpaddas *toString*-metod.

```
Nummer 1 - MoleTurtle
Nummer 2 - AbsentMindedTurtle (45% Frånvarande)
Nummer 3 - AbsentMindedTurtle (43% Frånvarande)
Nummer 4 - DizzyTurtle (Yrsel: 3)
Nummer 5 - MoleTurtle
Nummer 6 - MoleTurtle
Nummer 7 - AbsentMindedTurtle (71% Frånvarande)
Nummer 8 - DizzyTurtle (Yrsel: 5)
```

- Efter loppet ska första, andra och tredje plats skrivas ut (enligt exemplet nedan).

```
På plats 1: Nummer 5 - MoleTurtle
På plats 2: Nummer 6 - MoleTurtle
På plats 3: Nummer 4 - DizzyTurtle (Yrsel: 3)
```

## Checklista

I den här laborationen har du övat på att

- använda superklasser för att definera generella typer, och
- använda arv för att specialisera beteendet för specifika objekt.