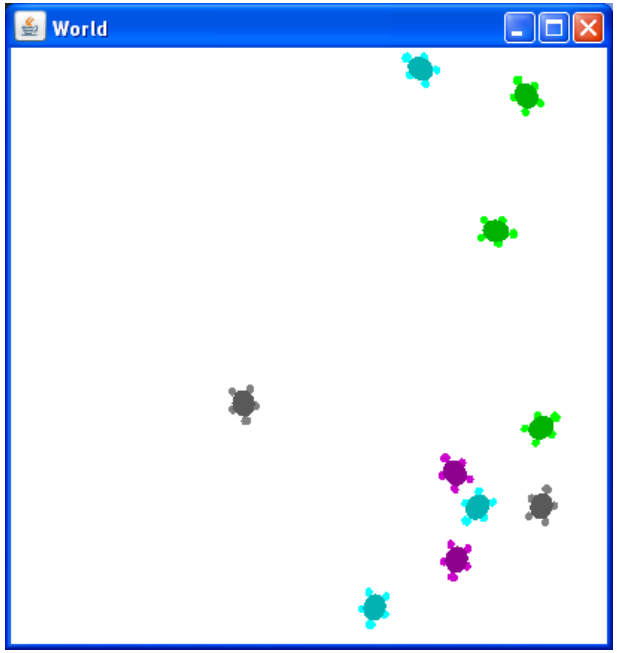
Introduktion till design av klasser

- Med en *klass* kan man skapa s.k. *objekt*.
- Ett objekt är en sorts variabel som refererar till ett antal andra variabler som beskriver själva objektet.
- Klassen beskrivs av pythonkod som är lagrat i en fil (python-filen).
- Vad är ett objekt?
- Jo, det refererar till värden (variabler) som är lagrade i datorns minne så länge objektet existerar



Vilka satser har programmet utfört för att få detta resultat?

Hur många synliga objekt har programmet skapat?

Hur många variabler har programmet skapat?

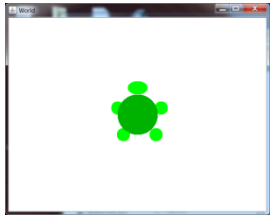
- När programmet som skapat objekten avslutas, försvinner objekten.
- Men klassen finns kvar, dvs pythonfilen vi körde för att skapa objekten.
- Vi kan dock spara objektens värden genom att programmet sparar dem, dvs skriver dem på en fil.

Klassen beskriver:

1. Vilka egenskaper som ett objekt kan ha – vilket beskrivs av klassens *instansvariabler**.
2. Hur kan man skapa objekt? Jo, det görs med en s.k. *konstruktor* i klassen
3. Vad kan man göra med ett objekt? Jo, med *metoderna* i klassen.

**De kallas *instans*variabler därför att när vi skapar ett objekt skapar vi en *instans* av en klass. Jmfr att vi har en pepparkaksform (mall) som vi skapar en pepparkaka av. Varje pepparkaka blir en instans av formen.

Om vi betraktar ett objekt som en modell för ett väsen/varelse/tingest, verklig eller fiktiv



kan vi säga att vi behöver metoderna för att kunna kommunicera med objekten:

- ändra objektet eller få veta något om objektet

När vi *designar* en klass X skriver vi ingen kod, utan börjar med att fundera på…

Vilka egenskaper skall ett objekt av klassen X ha?

Jmfr med klassen Turtle…

På vilka sätt skall man kunna skapa objekt av klassen X?

Jmfr med klassen Turtle…

Vad man kan tänkas vilja göra med (dvs hur kunna kommunicera) med ett objekt av klassen X?

Jmfr med klassen Turtle…

Därefter skissar vi på:

- **Egenskaperna**: Instansvariablernas namn och datatyp. Dessa kan vara av olika typer heltal, strängar, listor, objekt, ….
- **Skapandet:** Konstruktorer (med eller utan parametrar)
- **Kommunikationen:** Metodernas namn, returtyp och parametrar

  Två typer av metoder:

  mutators: Ändra objektet (jmfr metoden `forward` för en Turtle)

  accessors: Får veta något om objektet

  (jmfr metoden `distance(x, y)` för en Turtle)

Ett exempel på ett program https://www.greenfoot.org/scenarios/23533 som hanterar många klasser… (Torsten testkör det)