

DAT110 øving 5: Klasser og Objekter

Læringsmål

Du vil få erfaring med å bruke klasser og objekter. Du vil lære hvordan å lage en enkel klasse, og hvordan å lage og bruke objekter av denne klassen.

Denne øvingen inneholder to separate oppgaver, Slagskip og Sirkel. Du trenger bare å gjøre en av dem for å få godkjent øvingen. Hvis du trenger mengdetrening på klasser og objekter, så gjør du begge.

Oppgavebeskrivelse: Slagskip

Oppgaven består i å lage en enkelt klasse for et tenkt spill samt å lage kode som tester denne klassen og sjekker at den virker. Det er mulig at en enkel enspiller-utgave av hele spillet blir gitt som øving 8, og at en GUI-versjon blir gitt som øving 10.

Spillet Slagskip

Slagskipspillet er i utgangspunktet som følger:

- Det består av to brett, ett for hver spiller, på 10x10 ruter
- Hver spiller har 6 skip, ett med lengde 5, to med lengde 4, to med lengde 3 og ett med lengde 2
- Hver spiller skal plassere skipene sine ut på brettet som han vil. Skipene kan stå enten horisontalt eller vertikalt. Skipene kan ikke overlappe eller krysse hverandre.
- Deretter skal hver spiller prøve å skyte ned skipene til den andre spilleren. I sin tur sier spilleren hvilken rute han/hun skyter på. Motstanderen sier om han traff et skip og om skipet sank. En spiller kan kun skyte på samme rute en gang. Spilleren vet i utgangspunktet ikke hvor motstanderen har plassert sine skip.
- Et skip synker hvis alle rutene det ligger i er skutt på av motstanderen.
- En spiller taper spillet når alle skipene hans/hennes er senket. Den andre spilleren vinner.

Oppgaven

Oppgaven i øving 5 er å lage og teste klassen for et enkelt skip i spillet. Et skip har egenskapene `start_x`, `start_y`, `retning`, `lengde`, og antall treff skipet har tatt. Et skip skal ha følgende metoder:

- En konstruktør som lager skipet. Konstruktøren skal ta verdier av alle egenskapene unntatt antall treff som parametere. Du kan bruke `lengde 2` og `retning horisontal` som default-verdier. Antall treff skal settes lik 0.
- En `__str__` metode som lager og returnerer en streng som minimum inneholder startkoordinatene, lengden og antall treff skipet har tatt
- En metode `treff()` som registrerer at skipet har tatt en treff gjennom å øke antall treff egenskapen med 1
- En metode `er_senket` som returnerer `True` hvis skipet er senket og `False` hvis det fortsatt flyter.

Du skal lage denne klassen samt en `if __name__ == «__main__»:` blokk som tester den gjennom å lage minst to skip, med ulik lengde, og skyter på dem med `treff()` inntil de synker, men ikke lengre.

Oppgavebeskrivelse: Sirkel

Du skal lage og demonstrere bruken av en klasse for en sirkel. Du kan ta utgangspunkt i `Punkt` klassen fra videoene eller lage den fra bunnen av.

En sirkel har et senter og en radius. Senteret har x-koordinat og y-koordinat. Så `Sirkel`-klassen skal ha egenskapene `senter_x_koordinat`, `senter_y_koordinat`, og `radius`. `Sirkel`-klassen skal ha følgende metoder:

- `Flytt(delta_x, delta_y)` for å flytte sirkelen tilsvarende som for punkter
- `Areal()` for å beregne arealet til sirkelen. Arealet til en sirkel er $\pi \cdot r^2$ hvor r er radiusen.
- `Overlapper(annen_sirkel)` som sjekker om denne sirkelen og en annen sirkel gitt som parameter til metoden overlapper. To sirkler overlapper hvis avstanden mellom sentrene er lavere enn summen av radiene til de to sirklene.
- `Avstand` for å regne ut avstanden mellom to sirkler. Avstanden mellom to sirkler er lik avstanden mellom sentrene minus summen av radiene til begge sirklene. Hvis de to sirklene overlapper er avstanden 0.
- En `__str__` metode som returnerer en streng som inneholder koordinatene for senter og radius til sirkelen.
- **Frivillig:** `er_inni(annen_sirkel)`: Sjekker om den oppgitte sirkelen er helt inni denne sirkelen.
- **Frivillig:** `__eq__(annen_sirkel)`: Metoden `__eq__` er en Python spesialmetode som brukes til å sjekke om to objekter er like, `«==»` operatoren. Implementer en `__eq__` som sjekker at de to sirklene har samme senter og samme radius. Test denne ved å bruke `«==»` operatoren for å sjekke om ulike sirkler er like. Lag minst to sirkel-objekter som er like for denne testen.

`Overlapper` og `Avstand` kan skrives som funksjoner eller som metoder. Begge deler vil bli godkjent, men du må bruke riktig syntaks for den varianten du velger. For eksempel: `Overlapper` som funksjon tar to punkter som parametere. `Overlapper` som metode tar `self` og et annet punkt som parametere.

Skriv også en `if __name__ == «__main__»:` blokk som lager minst tre ulike sirkel-objekter med ulike radier, hvorav to overlapper hverandre mens den siste ikke gjør det. Kjør metodene `areal`, `overlapper` og `avstand`, skriv ut resultatene og se at det stemmer.

