

02101 Indledende Programmering 2021

Hjemmeopgave 2

Daniel Brasholt s214675
Jonathan Svarer Johansen s214671
Johannes Bjørn Nielsen s214682

Oktober 2021

Arbejdsdeling

Vi har i denne opgave valgt at dele hovedansvaret for de tre opgaver mellem os. Johannes har stået for opgave 1, Jonathan for opgave 2 og Daniel for opgave 3. Dog har vi naturligvis gennemlæst hinandens kode og sørget for, at kodestilen er så ensartet som muligt, at der ikke er ligegyldige linjer og at der ikke er tilfældige fejl.

Rapport

Problem 3

I opgave 3. ligger al koden i klassen *MovingPoint()*. Denne indeholder 2 constructors, hvor den ene konstruerer et *MovingPoint* med de inputkoordinater, hastighed og retning, som brugeren indtaster, og den anden blot laver et punkt med placering (0,0), retning 90° og hastighed 0. Sidstnævnte kalder da førstnævnte med de nævnte parametre. Derudover indeholder klassen følgende metoder:

- *setDirection(double direction)*. Denne sætter punktets direction til inputtet modulus 360 for at sørge for, at retningen altid er i intervallet $[0; 360[$. Derudover lægges retningen til 360, hvis den er negativ, så -30° bliver til $360 - 30 = 330^\circ$. Denne metode returnerer intet.
- *setSpeed(double speed)*. Ligesom *setDuration*, sætter denne hastigheden indenfor nogle grænser. Disse grænser implementeres ved, at *speed* bliver sat til maks-værdien af *speed* og 0, så den ikke bliver negativ, og min-værdien af *speed* og 20, så den ikke bliver over 20. Denne metode returnerer intet.
- *move(double duration)*. Denne metode flytter punktet som ønsket i opgaven. Dette gøres ved at tillægges punktets x-koordinat og y-koordinat følgende værdier:
 - x: Punktet tillægges punktets hastighed multipliceret med *duration* og cosinus til vinklen. Denne omregnes først fra grader til radianer, da *java.lang.Math.cos* anvendes, hvilket bruger radianer.
 - y: Som med x-koordinatet tillægges y punktets hastighed multipliceret med *duration* og sinus til vinklen. Også her omregnes først til radianer.

Denne metode returnerer intet.

- *turnBy(double angle)*. Denne metode lægger *angle* til punktets *direction*. Dette gøres gennem *setDirection*, så grænserne for vinklen bliver overholdt. Denne metode returnerer intet.
- *accelerateBy(double change)*. ud fra samme princip som *turnBy()*, ændres hastigheden gennem *setSpeed()*, så grænserne 0 og 20 for hastigheden bliver overholdt. Denne metoder returnerer intet.
- *toString()*. Denne metode returnerer en string, som er givet ud fra formatet $[x; y] d s$, hvor *d* er *direction* og *s* *speed*