Projekt Billetautomat

Af: Aksel og Jesper, 3.O

repo: <https://github.com/childstrangeler/billetautomaten>

(Det er Jesper der hedder childstrangeler)

## Indledning

Der er blevet givet en opgave ved navn “Projekt billetautomat”, opgaven omhandler at DSB mangler koden til deres billetautomater, de har en liste af krav der skal opfyldes og vil derudover gerne have programmørens tag på, hvad automaten mangler. DSB’s faste krav til automaten er som følger:

K1) Når en automat opstilles på en station skal installatøren kunne angive prisen per

billet. Efter installation skal prisen være fast.

K2) Automaten skal kunne fortælle kunden hvad en billet koster.

K3) Automaten skal kunne udskrive en billet mærket med DSB’s nye logo (i

terminalen).

K4) Automaten vil til enhver tid kunne vise kunden balancen, dvs. hvor mange penge

kunden har puttet i automaten.

K5) Ved opstilling skal installatøren kunne angive balancen, sådan at installatøren

kan udskrive en eller flere test-billetter uden at skulle putte penge i automaten.

K6) Medarbejderen skal kunne aflæse hvor mange billetter der er blevet solgt af

automaten og kunne tømme den for penge.

Derudover er DSB, som før nævnt, interesseret i at se hvad programmørerne selv kan tilføje af værdi til deres billetautomater.

## Vores tilgang til strukturering af user interface

Vi syntes det ville være nemmest at give brugeren en menu over mulige stop, og derefter lade brugeren vælge et start stop og en destination med en integer værdi. Her har brugeren dog også mulighed for at skrive “admin” i terminalen, for at forsøge at logge ind som administrator (her skal man dog kende det super secret password “admin1”). Som administrator har man mulighed for de værktøjer der skal bruges til installation af billetautomaten, samt udskrivning af gratis test billetter.

beskrive metoder og klasser

en klasse er et blueprint til at lave objekter, kan også ses som en brugs manual til hvordan man anvender et objekt, Java er et objektorienteret programmeringssprog, og det er derfor intuitivt at skulle benytte disse klasser til at gøre specifikke opdelte ting i programmet.

her er metoder så forskellige ting man kan gøre med det der objekt man med klassen har kreeret.

## Pathfinding, graph strukturer og dp

I vores DSB billetautomat cli, har vi valgt at bruge en breadth first- pathfinding algoritme, som taget en graf op finder den korteste rute (færrest nodes traversed).

Det gøres naivt, ved rekursivt at finde alle nodes med en bestemt afstand (depth) fra starts noden. Herefter tjekker man om destinations noden er her i blandt (så har vi fundet en path), eller bliver depth øget og så repeat.

Her har vi dog valgt at anvende memoization i stedet for en reel rekursiv funktion.

Dette er et paradigme inden for dynamic programming, der går ud på at definere rekursive funktioner, på en sådan måde at de kan huske information mellem kald, og dermed skære ned på runtime, ved at genanvende information.

Tit er det nemmere at definere sådan algoritmer uden egentlige rekursive kald, men bare forestille sig algorithmen som værende rekursiv. På den måde får man også en mindre call stack.