**Billetautomaten**

Du får til opgave at programmere en billetautomat – en meget naivbilletautomat (til at starte med).  
  
Vi ønsker at designe vores klasser med access modifiers, så hver instance af en klasse håndtere dets eget state gennem setter/getter methods. Når vi skriver vores klasser betyder det at dets attributes skal være **private**og dets methods samt constructor skal være **public**.

1. Lav en klasse med navnet **NaivBilletautomat**.
2. Denne klasse skal have tre attributes – alle af type **int**.  
   Den første skal holde på prisen for en billet i en automat (giv den variabelnavnet **pris**).  
   Den anden skal holde på det beløb, som løbende indsættes i automaten til en billet (giv den variabelnavnet **saldo**).  
   Den tredje skal holde på det totale beløb som maskinen har taget imod for dets billetter (giv den variabelnavnet **total**).
3. Lav nu en contructor til klassen, som giver **saldo** og **total**en startværdi på 0.  
   Klassens constructor modtager også et argument. Den skal derfor have et parameter, nemlig **billetpris** af datatypen **int**. **billetpris** tilskrives attribute **pris**.
4. Klassen skal også have nogle methods. Du skal skrive to getter methods, hvor den ene returnerer værdien af **pris**(kald den getPris()), og den anden returnerer værdien af **saldo** (kald den getSaldo()).  
     
   Husk, at de hver især returnerer en variabel af datatype **int**.
5. Du skal nu skrive en setter method (kald den **indbetal()**). Den returnerer ikke noget. Til gengæld tager den et argument (skriv som parameter **int kroner)**, som skal tilføjes til værdien af **saldo.**  
     
   På den måde holder automaten løbende styr på hvor mange penge, der er indsat til en billet.
6. Du skal nu skrive endnu en setter method (kald den **printBillet()**). Denne method tager ikke nogen argumenter. Til gengæld simulerer den en billetudskrift, som, i mangel af en rigtig billetautomat, udskrives til konsollen.  
     
   Det er helt op til dig hvordan din billet skal ’se ud’, men her er mit kodeeksempel som inspiration:

1. System.out.println("########################");

2. System.out.println(" /\\\_/\\");

3. System.out.println("( o.o )");

4. System.out.println(" > ^ <");

5. System.out.println("Billet: " + pris + " kr.");

6. System.out.println("########################");

Udover at simulere en billetudskrift, så skal printBillet() også ændre på værdien af **total**og **saldo**. **total** skal tilskrives summen af **total** og **saldo**, og **saldo** skal tilskrives værdien 0.  
  
På den måde kan en billetautomat holde styr på hvor mange penge, den har modtaget for de billetter den udsteder, og vi får også nulstillet saldoen så den er klar til næste gang nogen køber billet.

1. Lav en ny klasse og kald den **NaivBilletautoimatDemo**.
2. I denne klasse laver du en main method, hvori du laver en instance af **NaivBilletautomat**, og prøver dets forskellige metoder af for at se hvad vores billetautomat egentlig kan.  
     
   Hvilke beløb kan vi indsætte? Hvad sker der hvis vi indsætter mere end billettens pris? Mindre end billettens prisen?

NaivBilletautomat er ret passende for vores klasse, da de billetautomater vi kan lave ud fra den er meget naive og ikke særlig smarte. De tager glædeligt imod 0 kroners indbetalinger og man kan endda slippe afsted med at betale underpris for en billet!  
Lad os prøve at gøre vores klasse og dets fremtidige objekter lidtsmartere.

1. Ændre navnet af din klasse **NaivBilletautomat**til **Billetautomat**  
   (vha. Højre klik på klassen > Refactor > Rename… )
2. Lav et tjek (vha. if-else) i **indbetal()**.  
   Hvis **kroner**s værdi er større end 0, så tilføjes **kroner** til **saldo**, ellers så skrives der ud til konsollen, at man skal indsætte et postivt beløb, der er større end kroners værdi.
3. Lav et tjek i **printBillet()**.  
   Hvis **saldo**en er større eller lig med **pris**, så printer vi vores billet ud, hvis ikke, så skal **differencen** mellem **pris** og **saldo** udregnes, og der skrives ud i konsollen, at der skal indbetales mindst den pågældende diffference.

1. Ændre navnet af din klasse **NaivBilletautoumatDemo**til **BilletautomatDemo**.  
   Inde i klassens tilhørende main method, burde du allerede have en instance af **Billetautomat** (dette blev ændret med ændringen af klassenavnet), og  
   prøv at udforske hvad et billetautomat-objekt nu lan med vores små ændringer.

Nu har vi blot ved at lave et par tjek I vores methods, lavet et ‘klogere design’ af vores klasse og dets fremtidige objekter.  
  
Prøv at læs koden igennem, prøv at forstå præcis hvad der sker, og skriv gerne kommentarer over hver method og attribute, der fortæller præcis hvad de gør/bruges til.

1. Refactorer din kode, sådan at du opretter en ny Java Class der hedder Billet, der har en billetpris og en printBillet metode. Lav nu en MeneskeBillet og en CykelBillet der begge arver fra Billet class og sætter deres egen pris og printer deres egen billet.