

## Lektion 2

### Opgave 1

Gruppe nr. 15.

Navn og studienummer

#### Opgave 1:

Karl Emil Jeppesen	s180557
Alfred Röttger Rydahl	s160107
Said Abdullahi	s185013

#### Opgave 2:

Søren Poulsen	s180905
Rasmus Sander Larsen	s185097
Noah F. M. Hamza	s185084

### **Timeregnskab**

Navn	Timer
Karl Emil	1
Alfred	1
Saaïd	1
Søren	1
Rasmus	1
Noah	1

### **Kravspecifikation**

Denne opgave går ud på at designe, implementere og teste et program der kan udregne rumfanget af en cylinder ved input fra brugeren af højde og radius.

### **Design**

Det generelle design er at brugeren promptes til at indtaste et input.

Det testes om dette input er brugbart – det er et tal og større eller lig med nul.

Inputtet anvendes til at beregne **rumfanget af cylinderen** og værdien udskrives til brugeren.

Det søges at designe programmet således at det genstarter ved 'forkert' input.

## **Implementering**

Dette program gør kun nytte af klassen "Main.java", som indeholder hhv. metoderne main() og calcCylVol(). Vi har valgt at lave funktionen calcCylVol(), så koden bliver mere læsbar ved at beregningerne får sin egen funktion.

Vi gør brug af en Try Catch blok for at sikre programmet mod nedbrud, hvis brugeren skulle give et ugyldigt input. Vores Try Catch blok er derudover indkapslet i en "while" løkke, som gør at vi kan hoppe fra Catch til Try igen, så brugeren dermed får en ekstra chance. Variablerne "count" og "maxLoopCount" har som funktion at sikre, at vi ikke ender ud i en uendelig løkke.

Det beregnede rumfang bliver derudover afrundet til 2 decimaler ved brug af Java's indbyggede Math klasse.

## **Test**

### **Positive test**

Programmet blev testet efter det var skrevet og fungerede i forhold til positive tal, dog endte man med et tal med mange decimaler, da vi anvendte variabelen doubles. Dette blev lavet ved at anvende "Math.round()" til at forkerte outputtet ned til 2 decimaler.

### **Negative test**

Første iteration af programmet skelnede ikke mellem positive og negative tal. Dette gav muligheden for negative input og dermed output. Denne fejl blev rettet med en if/else statement med condition (input >= 0).

Programmet gav en InputMismatchException fejlmeddelelse ved input af bogstaver. Denne fejl blev rettet ved brug af en try-catch blok. Denne fanger alle input fejl som følge af indtastninger der ikke er tal.

Meget store og små tal håndteres udmærket af programmet da det scanner hele inputtet som en string og derefter konverterer til double. Det anses, at opløsningen i doubles er tilstrækkelig til formålet. Om nødvendigt kan mængden af decimaler ændres.

Det er med seneste version af programmet ikke lykkes at få det til at 'crashe'.