

02100 Indledende Programmering og Softwareteknologi - Gruppeaflevering 2

Mikkel s255468, Max s255964, Halfdan s256111

September 2025

1 Arbejdsdeling

Løsningsforslag blev diskuteret fælles hvorefter arbejdet blev uddeligeret. Max havde hovedansvaret for opgave 1 og 3, og Mikkel havde hovedansvaret for opgave 2 og 4. Halfdan gav sparring til de forskellige opgaver og refererede til sine egne løsningsforslag når der var brug for hjælp. Hvert program blev gennemgået og præsenteret af deres forfatter hvorefter vi diskuterede om løsningen gav mening.

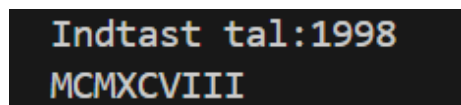
2 Opgave 1

Scanner der tjekker inputtet af værdi fra 1 til 3999. Her printes det romanske tal der findes ved at opdele inputtet i "små bider" med modulus og division. Da tallet er en Interger værdi undlades alle decimaler. Vi tjekker nu om tallet ligger inden for intervallerne 0-3, 4, 5-8 og 9 og returnere romersk tal for hvert input tal.

Tests:

Indtast tal:1998 (Se figur 1)

MCMXCVIII

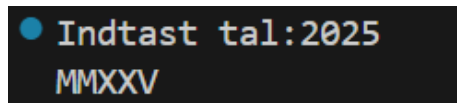


```
Indtast tal:1998
MCMXCVIII
```

Figur 1: Output 1998

Indtast tal:2025 (Se figur 2)

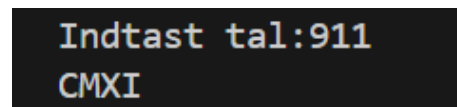
MMXXV

A terminal window with a dark background. It shows a blue prompt character followed by the text "Indtast tal:2025" on the first line and "MMXXV" on the second line.

```
● Indtast tal:2025
MMXXV
```

Figur 2: Output 2025

Indtast tal:911 (Se figur 3)
CMXI

A terminal window with a dark background. It shows the text "Indtast tal:911" on the first line and "CMXI" on the second line.

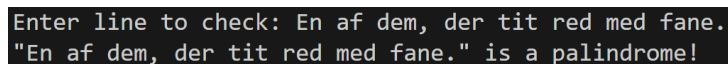
```
Indtast tal:911
CMXI
```

Figur 3: Output 911

3 Opgave 2

I vores endelige løsningsforslag starter vi med at tage input fra konsollen, ved at indføre en scanner, og gemme inputtet i en variable text af typen String. Derefter indfører vi en metode, der har returtypen boolean. Metoden kalder vi isPalindrome og den har vores variable text som dens parameter. metoden bryder teksten ned så man har en sammenhængene streng af små bogstaver. derefter indføres der en tom variable af typen String, som vi kalder s. vi kører derefter en if løkke, der iterer over længden af text og lægger hvert bogstav i text til den tomme streng s. til sidst returneres sandhedsværdien af om s er lig med text. i vores main metode laver vi så et if statement der tager vores isPalindrome metode som betingelse. hvis betingelsen er sand printer programmet, at den det givne input er et palindrom, ellers printer programmet, at det ikke er et palindrom. Som det fremgår at testene fungerer programmet på både ord, sætninger og tal, men det kan ikke klare hvis der er andre tegn end punktum, komma og mellemrum. Dette kunne løses, ved at ændre i den del af programmet der forkorter text strengen ned. Se Figur 4,5,6 og 7 for test af programmet.

Tests:

A terminal window with a dark background. It shows two lines of text: "Enter line to check: En af dem, der tit red med fane." on the first line and "\"En af dem, der tit red med fane.\" is a palindrome!" on the second line.

```
Enter line to check: En af dem, der tit red med fane.
"En af dem, der tit red med fane." is a palindrome!
```

Figur 4: Input: En af Dem, der tit red med fane.

```
Enter line to check: OTTO
"OTTO" is a palindrome!
```

Figur 5: Input: OTTO

```
Enter line to check: OTTO!
Is not a palindrome
```

Figur 6: Input: OTTO!

```
Enter line to check: 20.02.2002
"20.02.2002" is a palindrome!
```

Figur 7: Input: 20.02.2002

4 Opgave 3

Scanner der tjekker det indskrevne tal der kaldes iterations. Vi simulerer buffons needle ved brug af Random for bot (bottom) og for angle, vi tjekker derefter om succes ved top (toppen) er over/lig 2, denne findes ved at tage $\text{bot} + \sin(\text{angle})$. Vi tjekker teorien og tager iterations over successes og får nogle tal der ligger tæt på pi.

Tests:

Enter number of iterations: (Se figur 8)

10000000

10000000/3181208=3.143459968665991

```
Enter number of iterations:
10000000
10000000/3182412=3.142270705364359
```

Figur 8: Pi test 1

Enter number of iterations: (Se figur 9)

10000000

10000000/3182412=3.142270705364359

```
Enter number of iterations:
10000000
10000000/3181208=3.143459968665991
```

Figur 9: Pi test 2

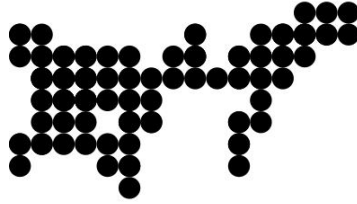
5 Opgave 4

Vi bruger en Scanner til at tage et input fra konsollen, hvilket vi gemmer i en variable `n` af typen `int`. For at sikre, at brugeren indtaster et heltal og ikke noget tekst, har vi implementeret en metode, som vi brugte til at løse opgave 4 på uge 4 ugeseddelen. Metoden står beskrevet i bogen[1, p. 357-358]. Den er navngivet `getInt` og prompter brugeren indtil de har indtastet et heltal. Dog virker programmet ikke, hvis brugeren indtaster et negativt heltal så hvis den indtastede værdi er mindre end 0 kaster vi en `IllegalArgumentException`. Derefter initialiserer vi variablerne `x` og `y`, som repræsenterer koordinaterne i det grid vi laver. Og variablen `steps` der tæller op hver gang der tages et skridt i gridet. Derefter kører vi en `while` løkke der tilfældigt ændrer `x` eller `y` med 1, så længe en af koordinaterne er mindre end gridets størrelse. Dette indtegnes i et grid af størrelsen $n \times n$ ved brug af `stdraw` biblioteket. Som det fremgår af testene (se figur 10, 11, 12, 13) virker programmet som ønsket på en række input.

Tests:

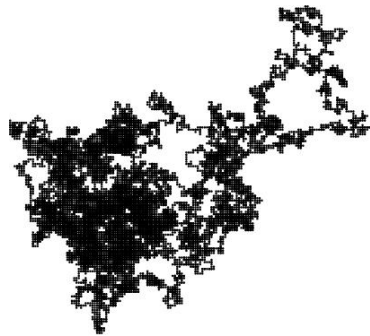
```
Enter grid size: et
Not an integer; try again.
Enter grid size: hundrede
Enter grid size: tre
Not an integer; try again.
Enter grid size: -10
Exception in thread "main" java.lang.IllegalArgumentException: Negative integer, Can't compute
```

Figur 10: Input: et, hundrede, tre, -10



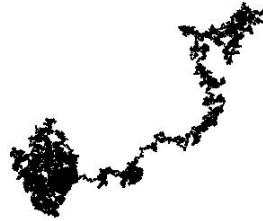
```
Enter grid size: 10  
Number of steps: 107
```

Figur 11: Input: 10



```
Enter grid size: 100  
Number of steps: 4245
```

Figur 12: Input: 100



```
Enter grid size: 1000  
Number of steps: 444659
```

Figur 13: Input: 1000

Litteratur

- [1] Reges Stuart, Stepp Marty, *Building Java Programs - Selected Chapters*, Pearson/DTU Compute, 2020.