

Inlämning 3

Fredrik Pettersson

June 2021

Introduktion

Detta är den tredje uppgiften i kursen 1TD317, Datorer och programmering. Det som skulle skapas var en simpel kalkylator som kan hantera tre olika beräkningar. Addition, multiplikation samt kvadratsumma. Programmet tar emot två tal samt val av beräkningsstyp. Därefter beräknas alla tal emellan det minsta till det högsta talet. Tex. talen 3 och 5 matas in samt addition väljes. Då blir resultatet 12 då $3+4+5=12$, eller om multiplikation väljes blir resultatet 60 då $3*4*5=60$. Åtta olika tester genomfördes och redovisas nedan.

Test 1

Första testet tar emot siffrorna 3 och 5 samt addition som beräkningsstyp. Resultatet blir 12 då $3+4+5=12$

```
Mata in ett tal
3
Valt tal: 3

Mata in ett tal
5
Valt tal: 5

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
a
Du valde addition som beräkningsstyp

Svaret är: 12

Process finished with exit code 0
```

Test 2

Samma som ovan fast med multiplikation istället $3*4*5=60$.

```
Mata in ett tal
█
Valt tal: 3

Mata in ett tal
█
Valt tal: 5

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
█
Du valde multiplikation som beräkningsstyp

Svaret är: 60
```

Test 3

Samma som ovan fast med kvadratroten istället $(3*3)+(4*4)+(5*5)=50$.

```
Mata in ett tal
█
Valt tal: 3

Mata in ett tal
█
Valt tal: 5

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
█
Du valde kvadratsumma som beräkningsstyp

Svaret är: 50

Process finished with exit code 0
```

Test 4

Multiplikation med talen -3 och 5 ger resultatet 0 då $(-3)*(-2)*(-1)*(0)*(1)*(2)*(3)*(4)*(5)=0$.

```
Mata in ett tal
3
Valt tal: -3

Mata in ett tal
5
Valt tal: 5

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
m
Du valde multiplikation som beräkningsstyp

Svaret är: 0

Process finished with exit code 0
```

Test 5

TVå tal matas in i "fel" ordning. Då måste talen sorteras vilket görs i funktionen sortera. 5 och 3 ger resultatet 12 då $3+4+5=12$.

```
Mata in ett tal
5
Valt tal: 5

Mata in ett tal
3
Valt tal: 3

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
a
Du valde addition som beräkningsstyp

Svaret är: 12
```

Test 6

Two negative numbers are entered: -3 and -5, and multiplication is chosen. The result is -12, since $(-5) + (-4) + (-3) = -12$.

```
Mata in ett tal
3
Valt tal: -3

Mata in ett tal
5
Valt tal: -5

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
m
Du valde addition som beräkningsstyp

Svaret är: -12

Process finished with exit code 0
```

Test 7

The numbers 1 to 100 are entered with the square root as the calculation type. The result is 338350.

```
Mata in ett tal
1
Valt tal: 1

Mata in ett tal
100
Valt tal: 100

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
k
Du valde kvadratsumma som beräkningsstyp

Svaret är: 338350

Process finished with exit code 0
```

Test 8

Test 8 vilket är en frivillig uppgift. Talen 1 och 1000 matas in med multiplikation som beräkningsstyp, detta kan även skrivas som 1000! eller 1000 fakultet. Detta resulterar i ett enormt stort tal, vilket variabeltypen integer inte klarar av, därav svaret 0.

```
Mata in ett tal
1
Valt tal: 1

Mata in ett tal
1000
Valt tal: 1000

Skriv in vilken beräkning du vill använda
a för addition, m för multiplikation eller k för kvadratsumman
m
Du valde multiplikation som beräkningsstyp

Svaret är: 0

Process finished with exit code 0
```

Beskrivning av koden

Detta problem går ut på att öva användandet av funktioner. Då man istället för att lägga all logik i main funktionen bryter upp problemet i flera mindre delar som löser en viss sak. Det enda main funktionen har till uppgift är att kalla på dessa funktioner i rätt ordning.

Användandet av funktioner har många fördelar. Framförallt så blir koden enklare att förstå. Varje metod ansvarar för en enda sak och koden blir då inte rörig. Det gör även koden enkel att justera då man bara behöver ändra små saker i en funktion istället för stora delar i koden.