

Kapitel 4 Loopar

Övning 2: Repeat...End-loopen

I denna tredje aktivitet för kapitel 4 kommer du att lära dig om **Repeat...End**-loopen. Vi jämför den med **For...** loopen och visar också varför den är mer kraftfull.

The Repeat... End Loop

Repeat...End-loopen kommer att fortsätta "loopandet" så länge som <villkor> är Falskt. Detta är exakt motsatt uppförandet för **While**-loopen men det är inte den enda skillnaden! Det ser ut så här, vilket påminner väldigt mycket om **While**-strukturen:

```
Repeat <villkor>
  <loopkropp>
End
```

Programen som visas till höger ger samma resultat vid körning. Vilka är skillnaderna?

<Villkor> är ett logiskt uttryck, t.ex. $X > 0$.

<loopkropp> är en uppsättning satser inklusive andra loopar och **If**-strukturer. Den **processas en gång** och fortsätter sedan **tills** <villkor> är sant. Alltså så här:

```
Repeat
  <loopkropp>
  tills <villkor> är sant End
```

Men nu finns det inget "tills" i TI Basic. Det är underförstått. Även om <villkor> är sant från start så processas fortfarande loopkroppen en gång eftersom villkoret testas i slutet av loopen.

I en **While**-loop är det en smart idé att ha initialvärden för variabeln eller variablerna. I en **Repeat**-loop sker detta inne i själva loopkroppen så där är det inte nödvändigt.

Precis som med **While**, ska det någonstans i loopkroppen finnas en programsats som påverkar <villkor> så att loopen eventuellt kommer att avslutas och satser efter loopen kan processas. Vanligtvis finns denna sats i slutet av loopkroppen.

Syfte:

- Lära sig strukturen och logiken hos **Repeat...End**-loopen.
- Jämföra den med **While...End**-loopen.
- Se hur **Repeat...End** används i programmering av Fibonaccis talföljd.

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: REPEAT
:0→A
:Repeat A>10
:Disp A
:A+1→A
:End
:
```

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: WHILE
:0→K
:While K≤10
:Disp K
:K+1→K
:End
:■
```

Programmera Fibonacci talföljd

Vi ska nu skriva ett program som visar talen i *Fibonacci* talföljd upp till ett speciellt värde. I denna talserie är varje tal summan av de två föregående. Det finns massor att läsa om denna välkända och intressanta talföljd på webben.

Till höger visas resultatet av en programkörning. Kan du skriva koden för ett sådant program utan att kika längre ner på sidan?

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: FIBONACC
N=?5
1
1
2
3
5
8
..... Klar
```

Vi börjar med en **Prompt**-sats för att få värdet för en övre gräns från användaren. De två första Fibonaccitalen är 1 och 1 så vi lagrar dessa värden i variablerna **A** and **B**. Dessa variabler kommer att användas för att beräkna de följande talen i talserien (upp till **N**).

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: FIBONACC
:Prompt N
:1→A
:1→B
:█
```

Sedan börjar vi **Repeat**-loopen med villkoret **A>N**, som betyder "tills **A** är större än **N**". I loopen så visar vi först de nuvarande två talen.

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: FIBONACC
:Prompt N
:1→A
:1→B
:Repeat A>N
:Disp A,B
:
```

Slutligen så beräknar vi de följande två Fibonaccitalen och avslutar loopen med **End**.

Dessa två sista satser visar att $A+B$ lagras i både **A** och **B** och det verkar då som **A** och **B** får samma värde. **Så är det absolut inte!** Prova med 1 och 1. Den *första* satsen lagrar $1+1$, som blir 2, i **A**. Den *andra* satsen lagrar $2+1$, som blir 3, i **B**.

Kör programmet med olika värden på **N**. Uppträder programmet som du tänkt dig?

```
NORMAL FLYT AUTO REELL RAD MP
PROGRAM: FIBONACC
:Prompt N
:1→A
:1→B
:Repeat A>N
:Disp A,B
:A+B→A
:A+B→B
:End
:
```

Ändra nu **Repeat**-villkoret så att det istället är **B>N**. Vad blir då resultatet? Hur kan man ändra programmet så att det visar en korrekt talserie och där det stannar exakt när det största Fibonaccitalet som är mindre än **N** har visats?