



Remarks

- The loop variable **increases** with 1 with every execution.
- The loop variable is **ordinal** – integer, character or logic types.
- **LowerBoundary**: is where the loop variable **starts**.
- **UpperBoundary**: is where the loop variable **stops**.
- The lower boundary must always be **smaller than** (or **equal to**) the upper boundary. The loop in Example 1 is executed as long as Down ≤ 5 .
- The loop variable (**A**) must also be **declared** using var.
- **The loop can be used to count backwards:**
For <LoopVariable> := UpperBoundary **downto** LowerBoundary **do**

The loop variable **decreases** with 1 with each execution of the loop.

UpperBoundary is where the loop **starts**.

LowerBoundary is where the loop **stops**.

The loop is executed as long as UpperBoundary \geq LowerBoundary.

Activity 6.1



Individual work

Write programs to do the following. The lower- and upper boundaries are chosen in two SpinEdit-components where integers are used as loop variables, else edit-components are used for input. Display the following in a listbox-component:

1. Integers from 1 to 10.
2. Integers from -10 to $+10$.
3. Letters from 'A' to 'D'.
4. Lowercase letters from 'd' to 'j'.
5. Values from False to True.
6. Values from True to False.
(Test whether all the examples can be executed!).



Group work

Determine the number of repetitions of each loop

1. For K := -6 to 6 do
2. For L := 20 downto 14 do
3. For M := False to True do
4. For J := True to False do
5. For K := 1 downto -4 do
6. For X := 'p' to 's' do
7. For J := True downto False do
8. For K := 5 downto 15 do