ŠKODA AUTO University

Computer Simulation of Logistics Processes

Programming in SimTalk

Jan Fábry

10/04/2023

Programming in SimTalk

Aim of the lecture

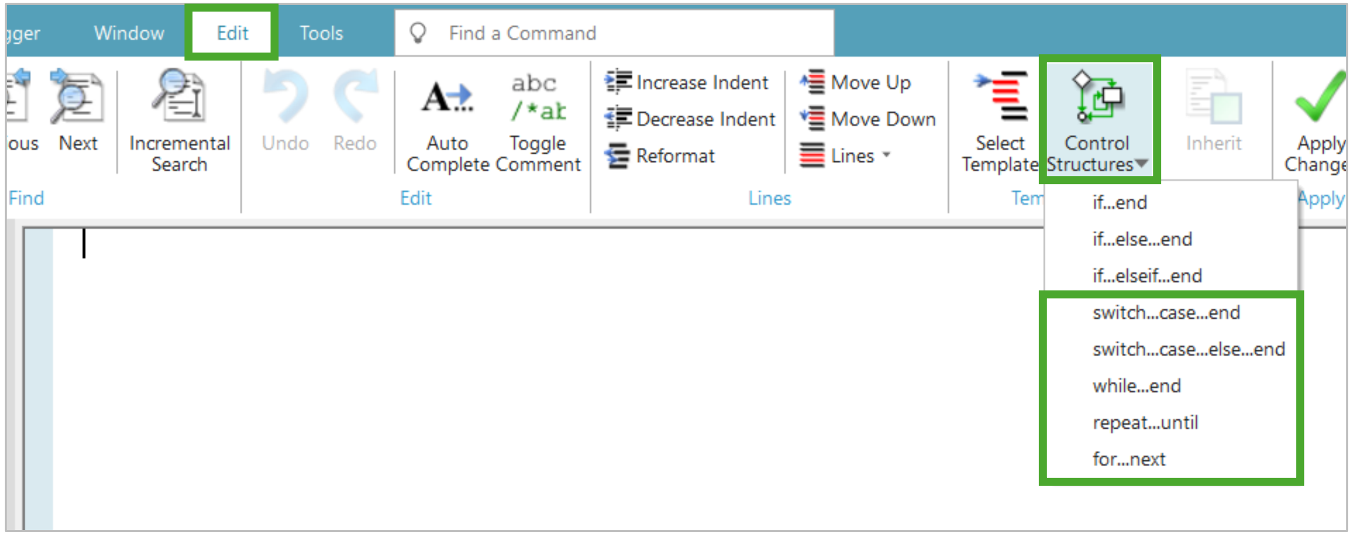
- To introduce the possibilities to work with cycles and tables.

Programming in SimTalk

ŠKODA AUTO University

Structure of the lecture

- Conditional commands with SWITCH:
  - switch … case … end
  - switch … case … else … end
- Commands with repeated run:
  - while … end
  - repeat … until
  - for … next
- Tables.



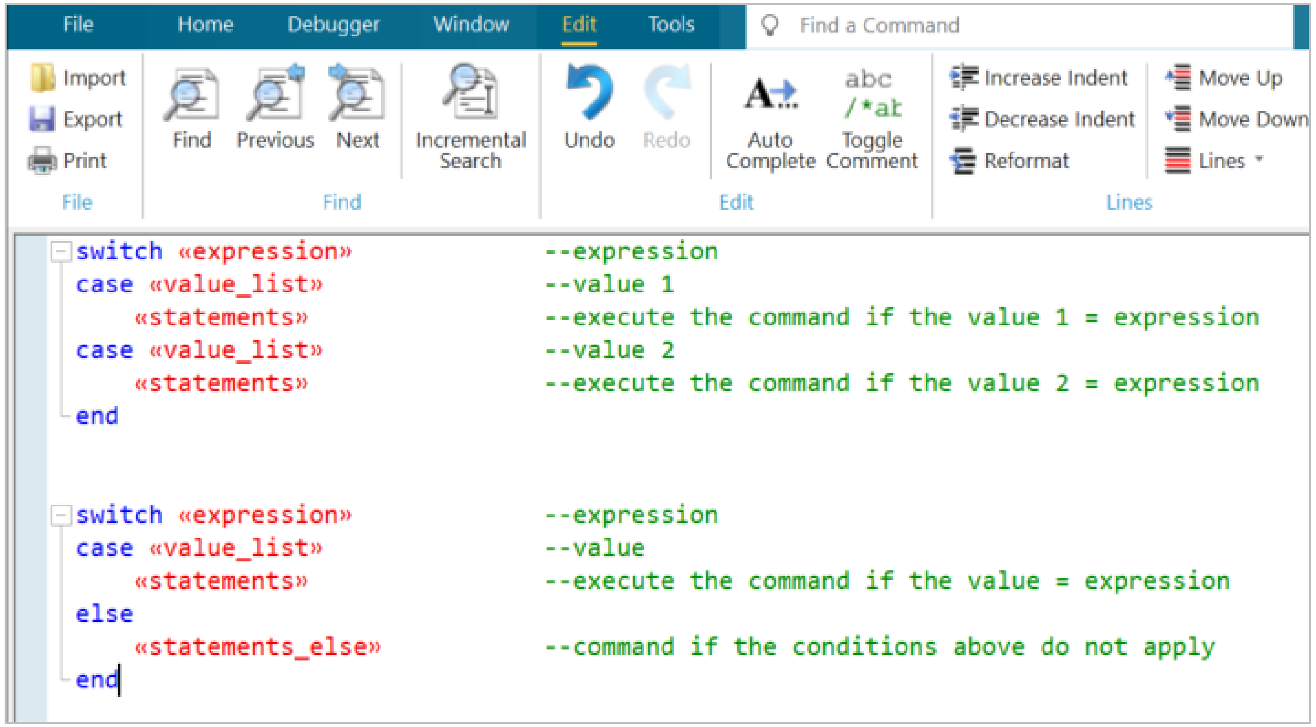
Programming in SimTalk

Conditional commands with switch

- Types:

  -
    ```
    switch <expression>
        case <value>
            <general command>
        case <value>
            <general command>
    end
    ```

  -
    ```
    switch <expression>
        case <value>
            <general command>
        else
            <general command>
    end
    ```



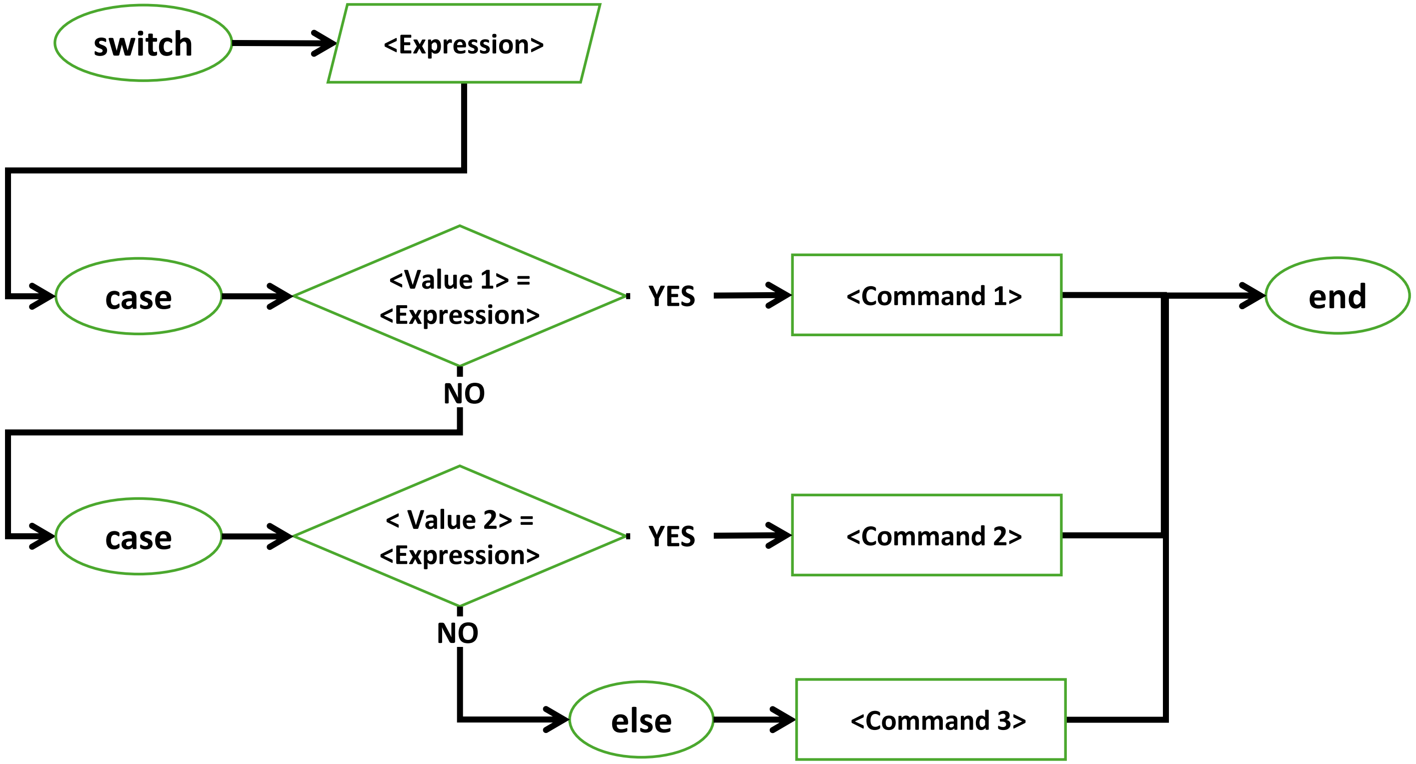
```
switch «expression»                --expression
case «value_list»                  --value 1
    «statements»                   --execute the command if the value 1 = expression
case «value_list»                  --value 2
    «statements»                   --execute the command if the value 2 = expression
end


switch «expression»                --expression
case «value_list»                  --value
    «statements»                   --execute the command if the value = expression
else
    «statements_else»              --command if the conditions above do not apply
end
```

- Syntax “case <value> <general command>” can be used multiple times.
- Value/expression can be integer, real or string data type.

Programming in SimTalk

Conditional commands with switch

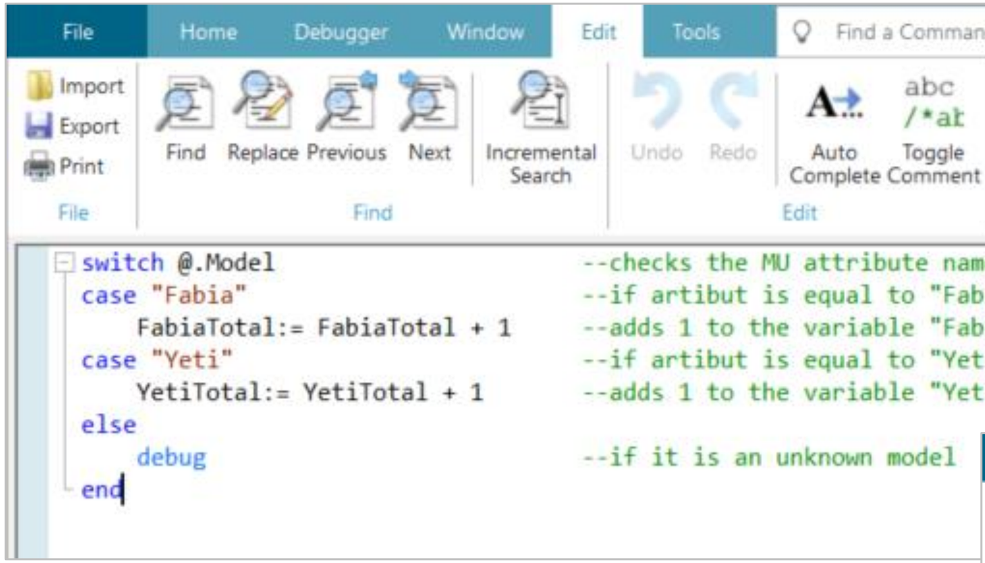
- Syntax diagram:



Programming in SimTalk

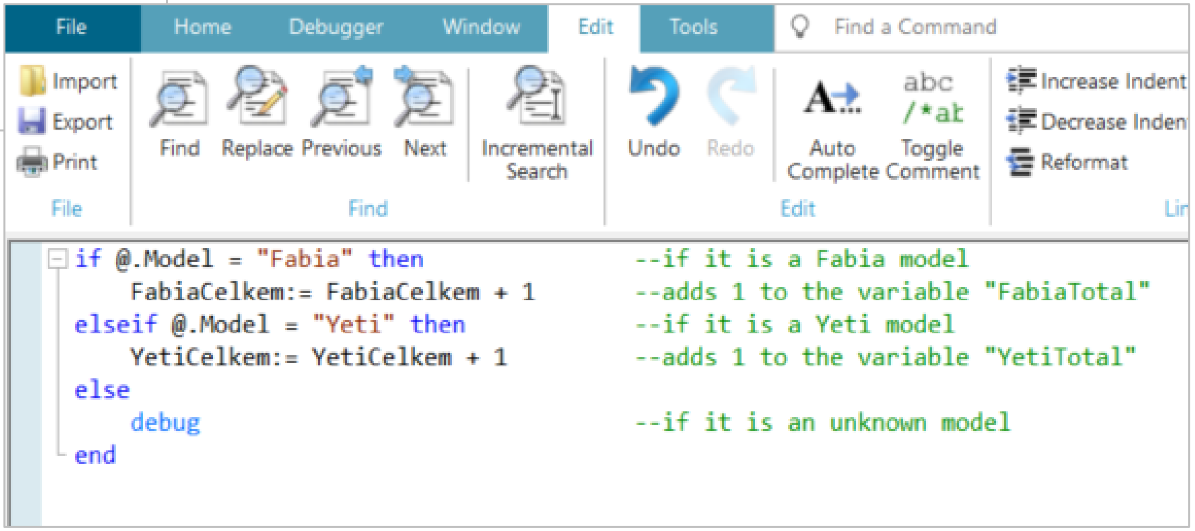
Conditional commands with switch

- Example:



```
switch @.Model                    --checks the MU attribute named model
case "Fabia"                      --if artibut is equal to "Fabia"
    FabiaTotal:= FabiaTotal + 1   --adds 1 to the variable "FabiaTotal"
case "Yeti"                       --if artibut is equal to "Yeti"
    YetiTotal:= YetiTotal + 1     --adds 1 to the variable "YetiTotal"
else
    debug                         --if it is an unknown model
end
```

- Example of using the IF command:

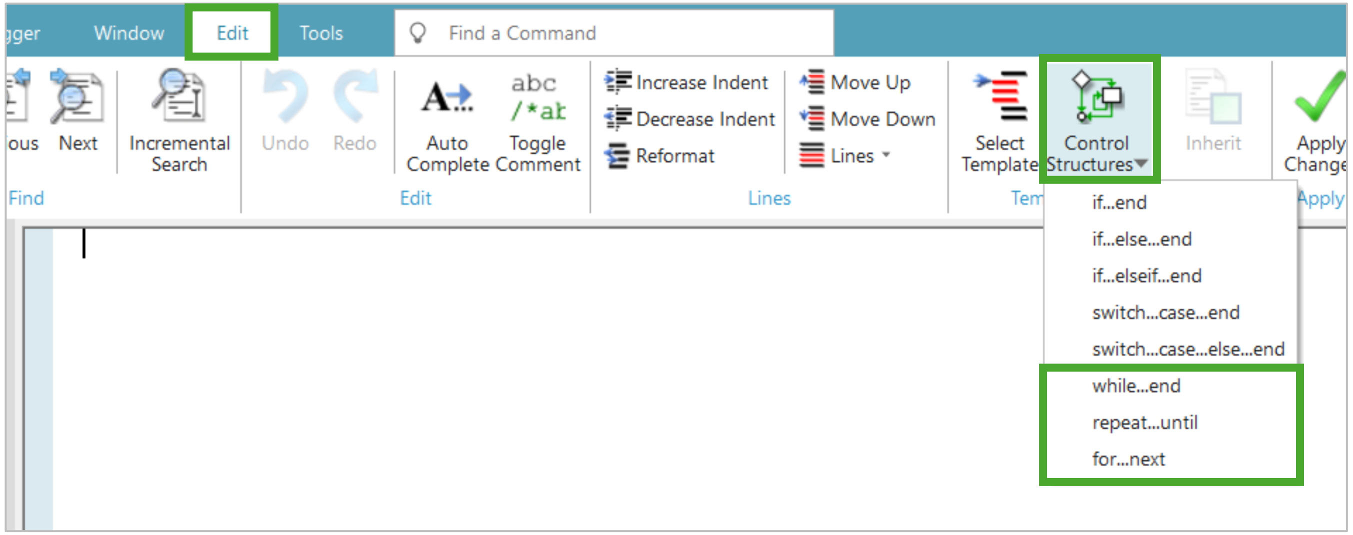


```
if @.Model = "Fabia" then             --if it is a Fabia model
    FabiaCelkem:= FabiaCelkem + 1     --adds 1 to the variable "FabiaTotal"
elseif @.Model = "Yeti" then          --if it is a Yeti model
    YetiCelkem:= YetiCelkem + 1       --adds 1 to the variable "YetiTotal"
else
    debug                             --if it is an unknown model
end
```

Programming in SimTalk

Commands with repeated run

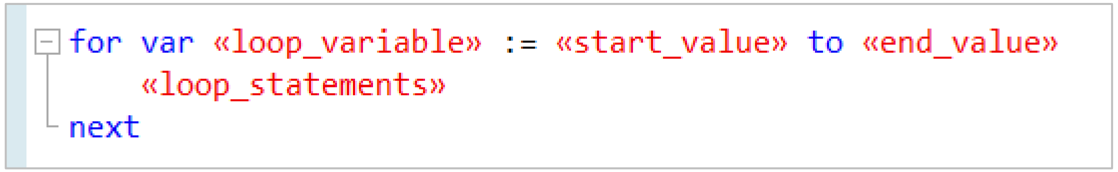
- The cycle is defined via control variable.
- The commands contained in the cycle are processed as long as the control variable does not exceed a final value, a certain condition is met or in the opposite, until a certain condition occurs.
- Basic types of cycles:
  - while … end
  - repeat … until
  - for … next



Programming in SimTalk

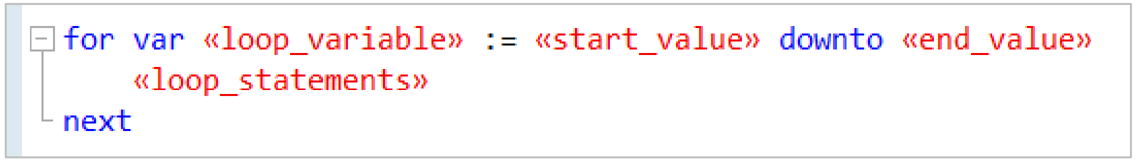
Commands with repeated run with for...next

- Types:
    - for <control variable := start value> to <end value>
      <Command sequence in the loop>
      end
        - initial value < final value



```
for var «loop_variable» := «start_value» to «end_value»
    «loop_statements»
next
```

- for <control variable := start value> downto <end value>
  <Command sequence in the loop>
  end
    - initial value > final value

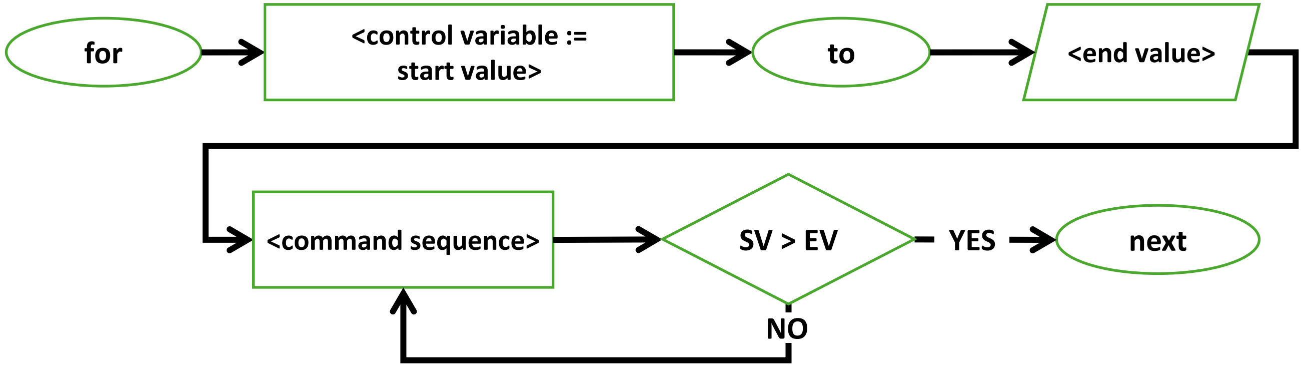


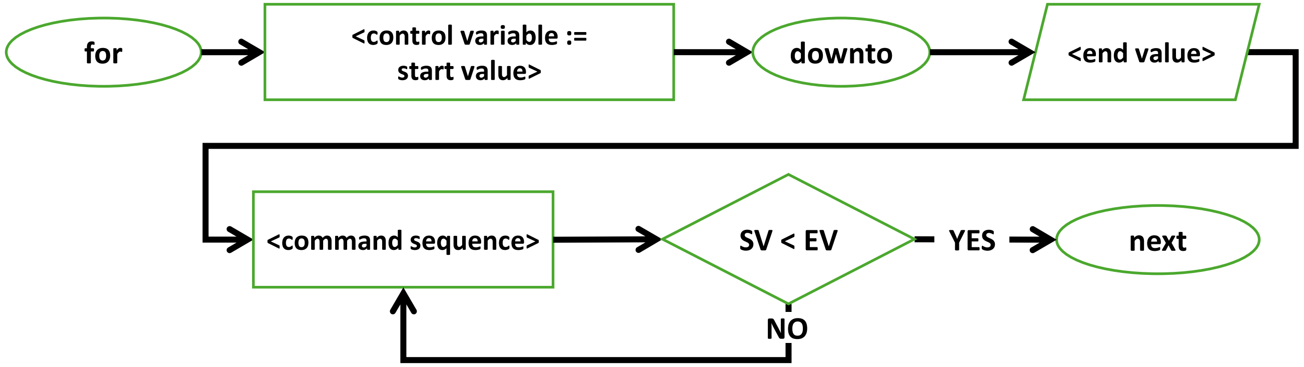
```
for var «loop_variable» := «start_value» downto «end_value»
    «loop_statements»
next
```

Programming in SimTalk

Commands with repeated run with for...next

- Syntax diagram:

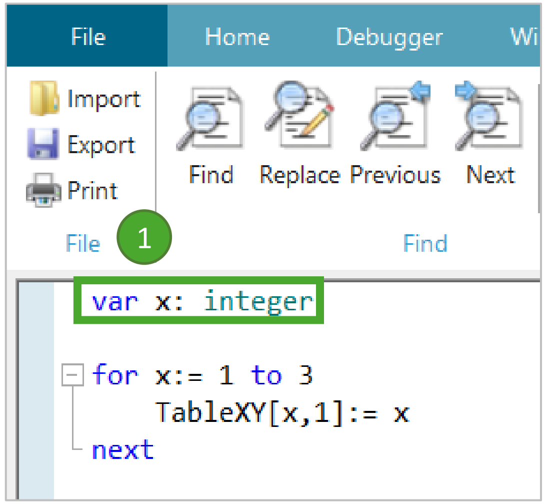




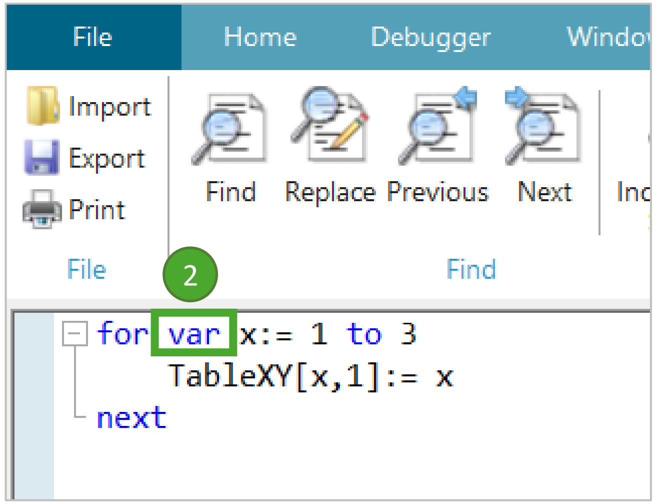
Programming in SimTalk

Commands with repeated run with for...next

- Definition of the control variable (loop_variable):
    - Control variable must be of data type integer.
    - It can be defined as local variable (1) or directly in the command via keyword “var” (2).
    - By using “var” in the command, the control variable becomes visible only in the loop.



```
var x: integer

for x:= 1 to 3
    TableXY[x,1]:= x
next
```

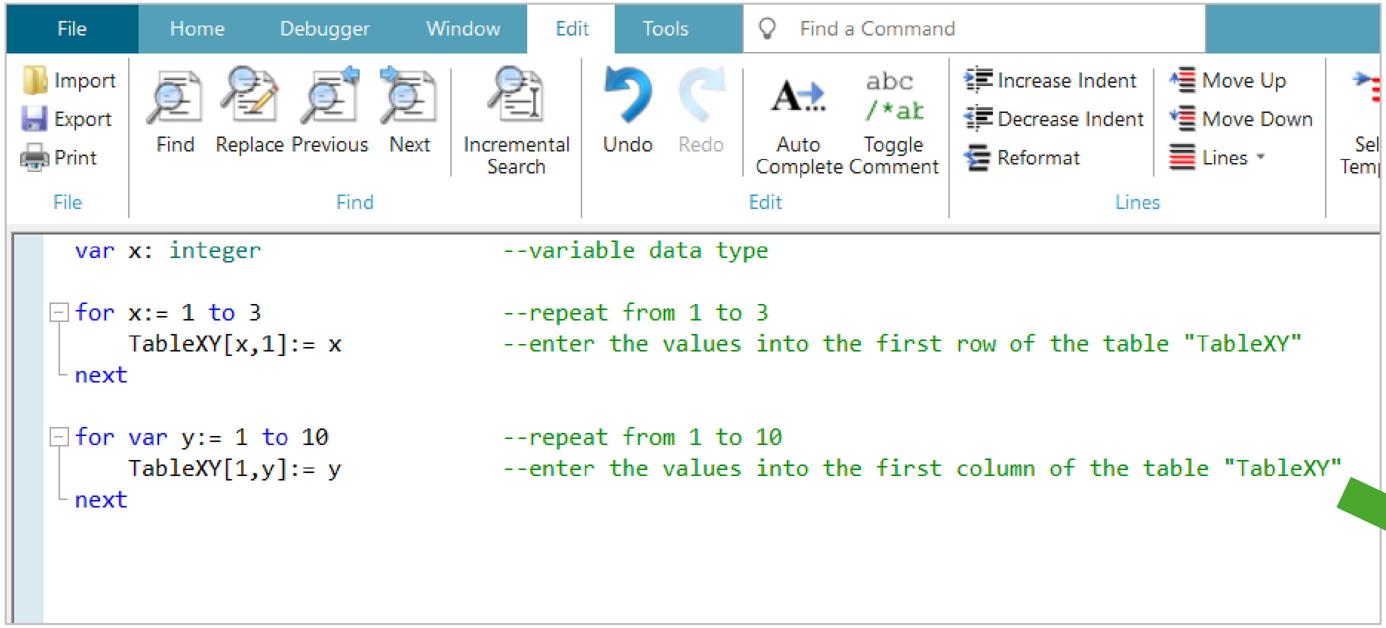


```
for var x:= 1 to 3
    TableXY[x,1]:= x
next
```

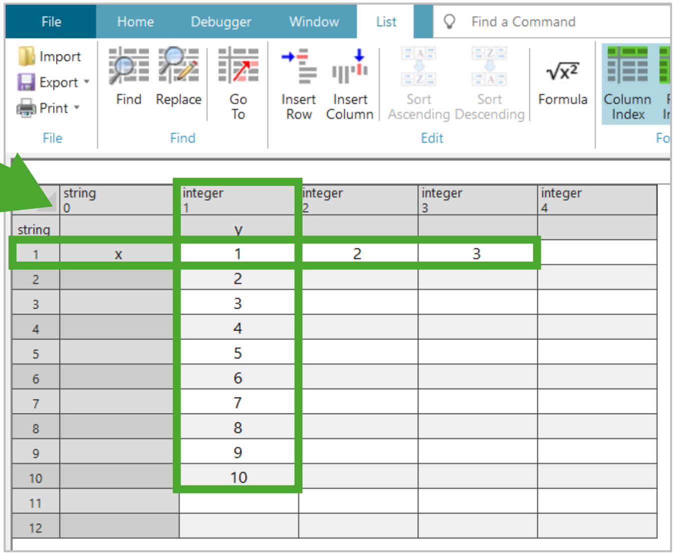
Programming in SimTalk

Commands with repeated run with for...next

- Example:



```
var x: integer                  --variable data type

for x:= 1 to 3                  --repeat from 1 to 3
    TableXY[x,1]:= x            --enter the values into the first row of the table "TableXY"
next

for var y:= 1 to 10             --repeat from 1 to 10
    TableXY[1,y]:= y            --enter the values into the first column of the table "TableXY"
next
```

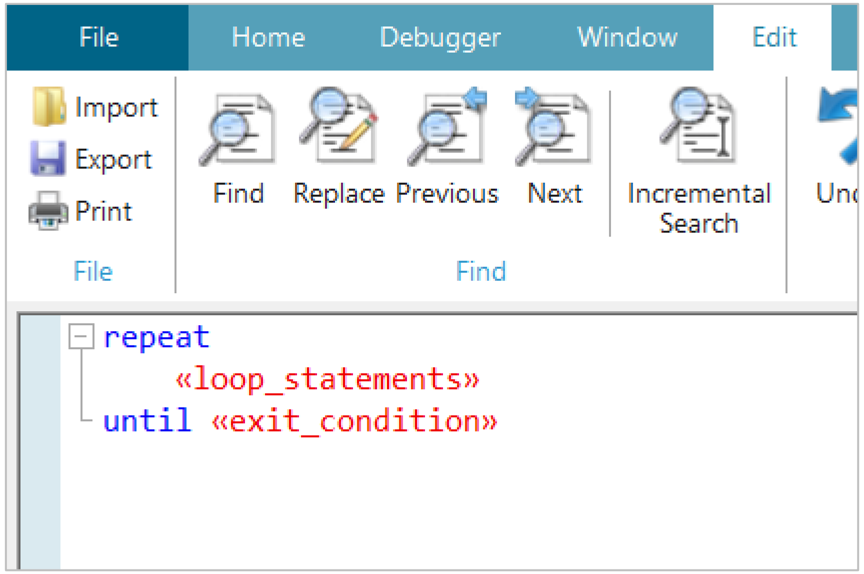


	string 0	integer 1	integer 2	integer 3	integer 4
string		y			
1	x	1	2	3	
2		2			
3		3			
4		4			
5		5			
6		6			
7		7			
8		8			
9		9			
10		10			
11					
12					

Programming in SimTalk

Commands with repeated run with repeat…until

- repeat
      <Command sequence in the loop>
  until <Condition for loop termination >



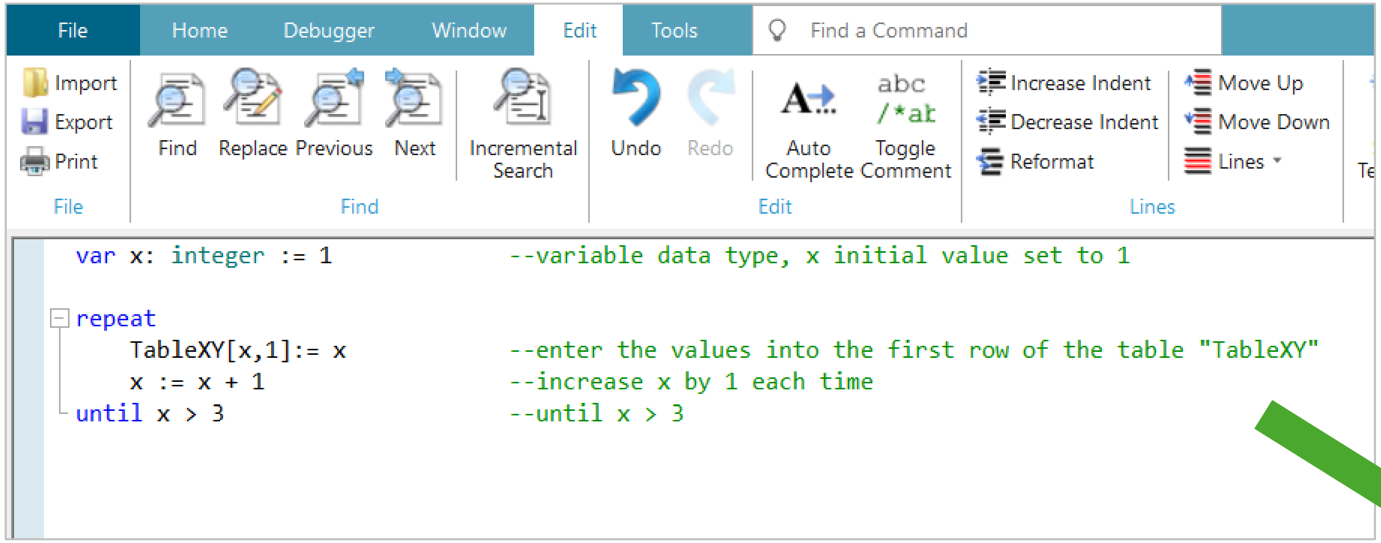
```
repeat
    «loop_statements»
until «exit_condition»
```

Programming in SimTalk

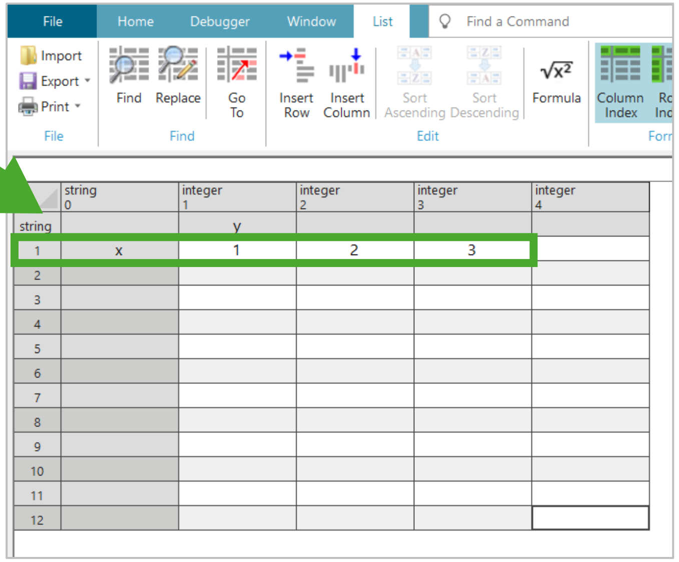
ŠKODA AUTO University

Commands with repeated run with repeat…until

- Example:



```
var x: integer := 1          --variable data type, x initial value set to 1

repeat
    TableXY[x,1]:= x         --enter the values into the first row of the table "TableXY"
    x := x + 1               --increase x by 1 each time
until x > 3                  --until x > 3
```

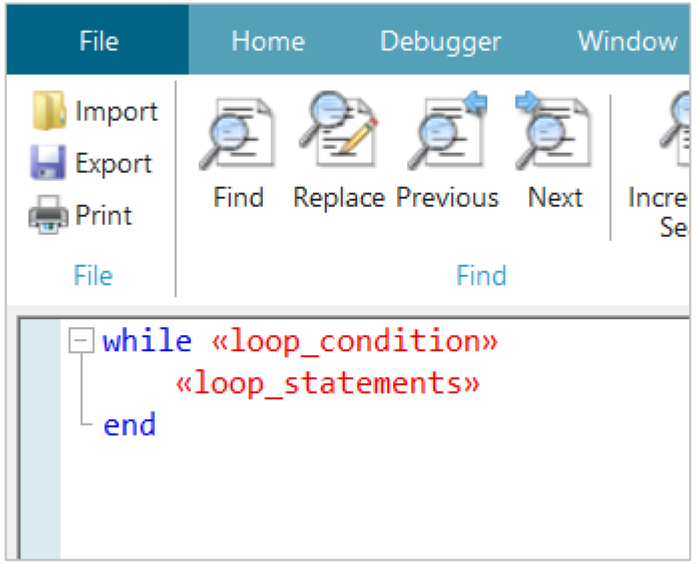


	string 0	integer 1	integer 2	integer 3	integer 4
string		y			
1	x	1	2	3	
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Programming in SimTalk

Commands with repeated run with while…end

```
while <Condition for the loop run>
        < Command sequence in the loop >
end
```



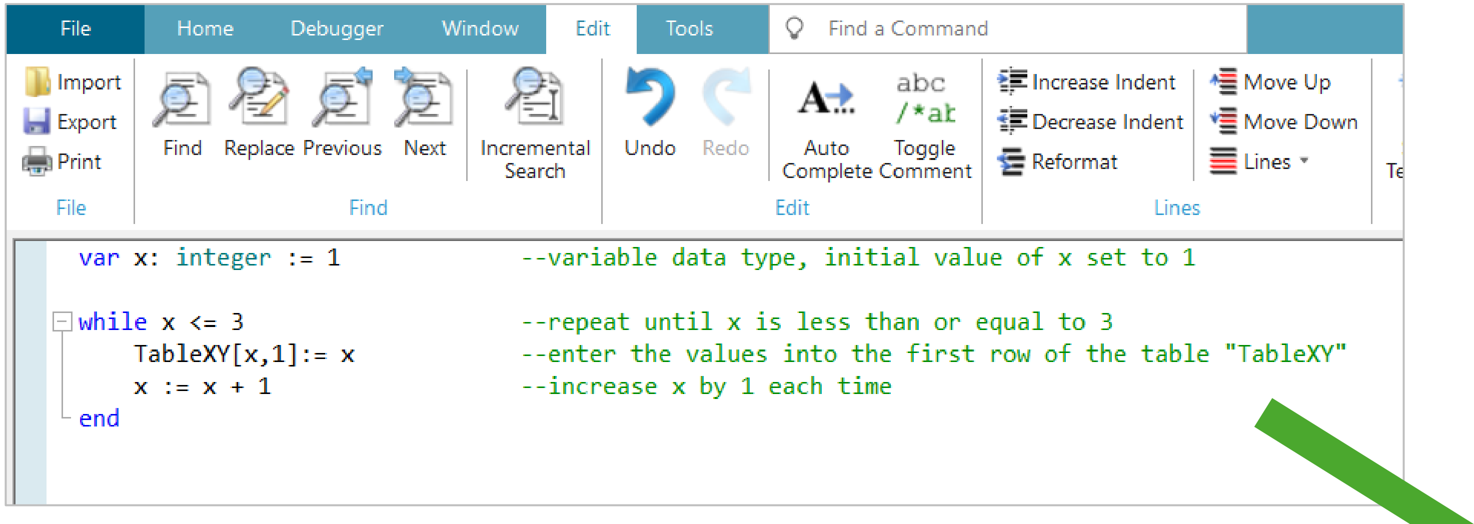
```
while «loop_condition»
    «loop_statements»
end
```

Programming in SimTalk

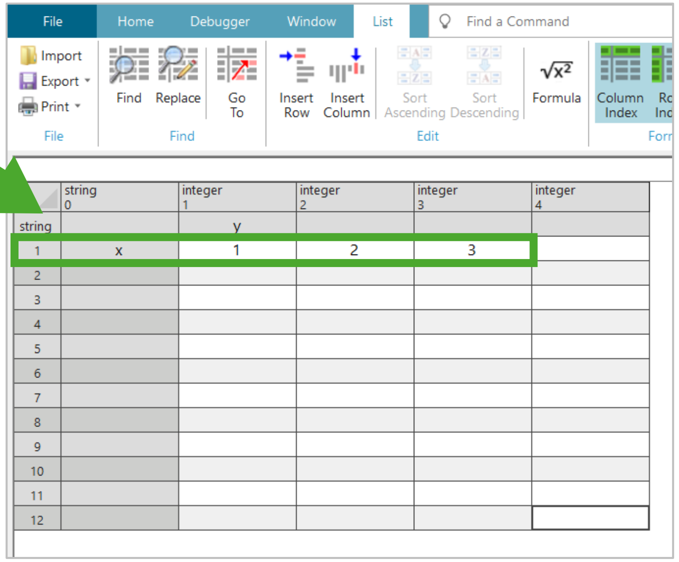
ŠKODA AUTO University

Commands with repeated run with while…end

- Example:



```
var x: integer := 1           --variable data type, initial value of x set to 1

while x <= 3                  --repeat until x is less than or equal to 3
    TableXY[x,1]:= x          --enter the values into the first row of the table "TableXY"
    x := x + 1                --increase x by 1 each time
end
```

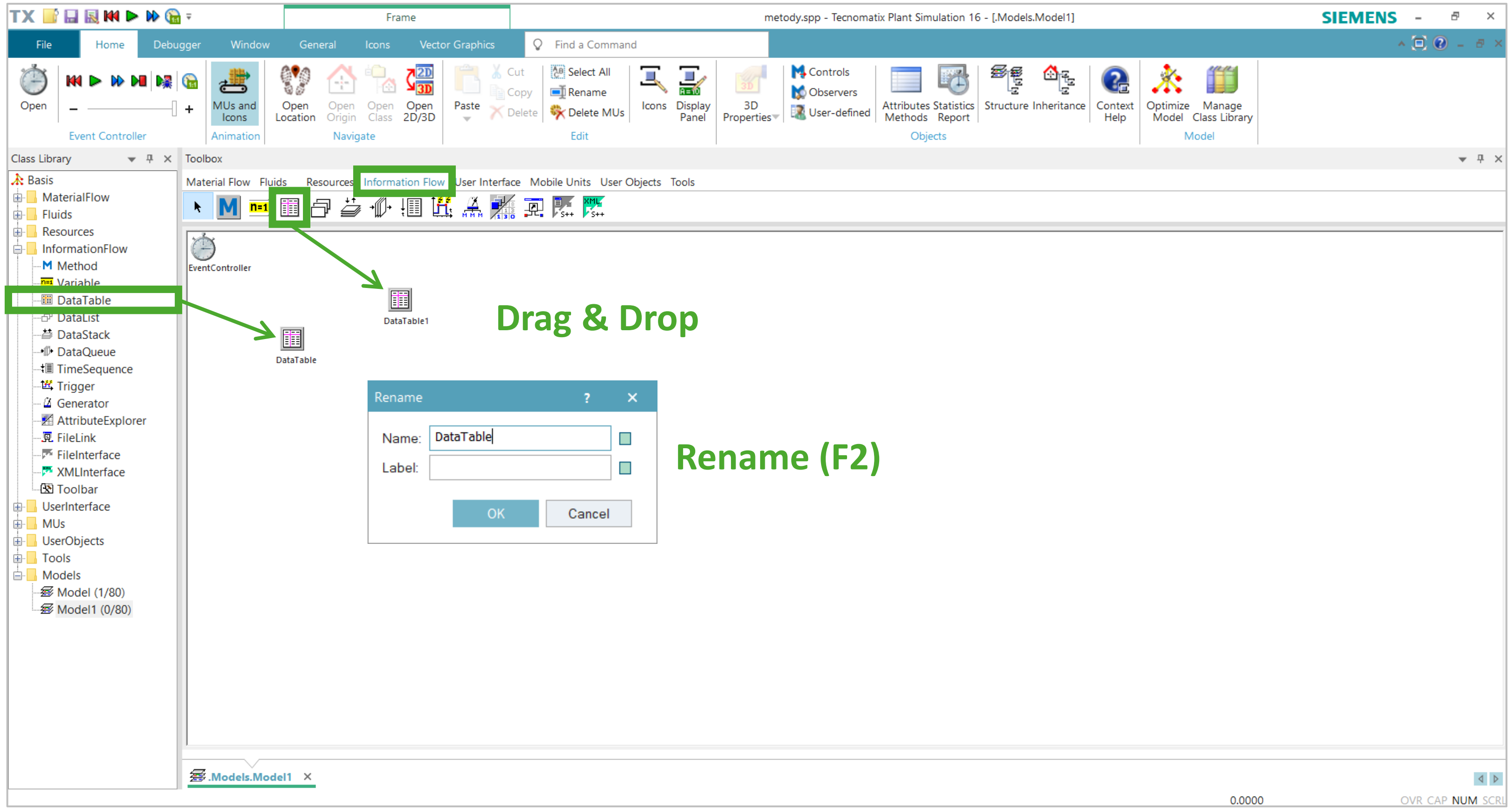


	string 0	integer 1	integer 2	integer 3	integer 4
string		y			
1	x	1	2	3	
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Programming in SimTalk

Table

- Table is an object composed of rows and columns.
- It is used for writing or reading values/expressions.
- Values/expressions can be added or deleted during the simulation run.



Drag & Drop

Rename

Name: DataTable

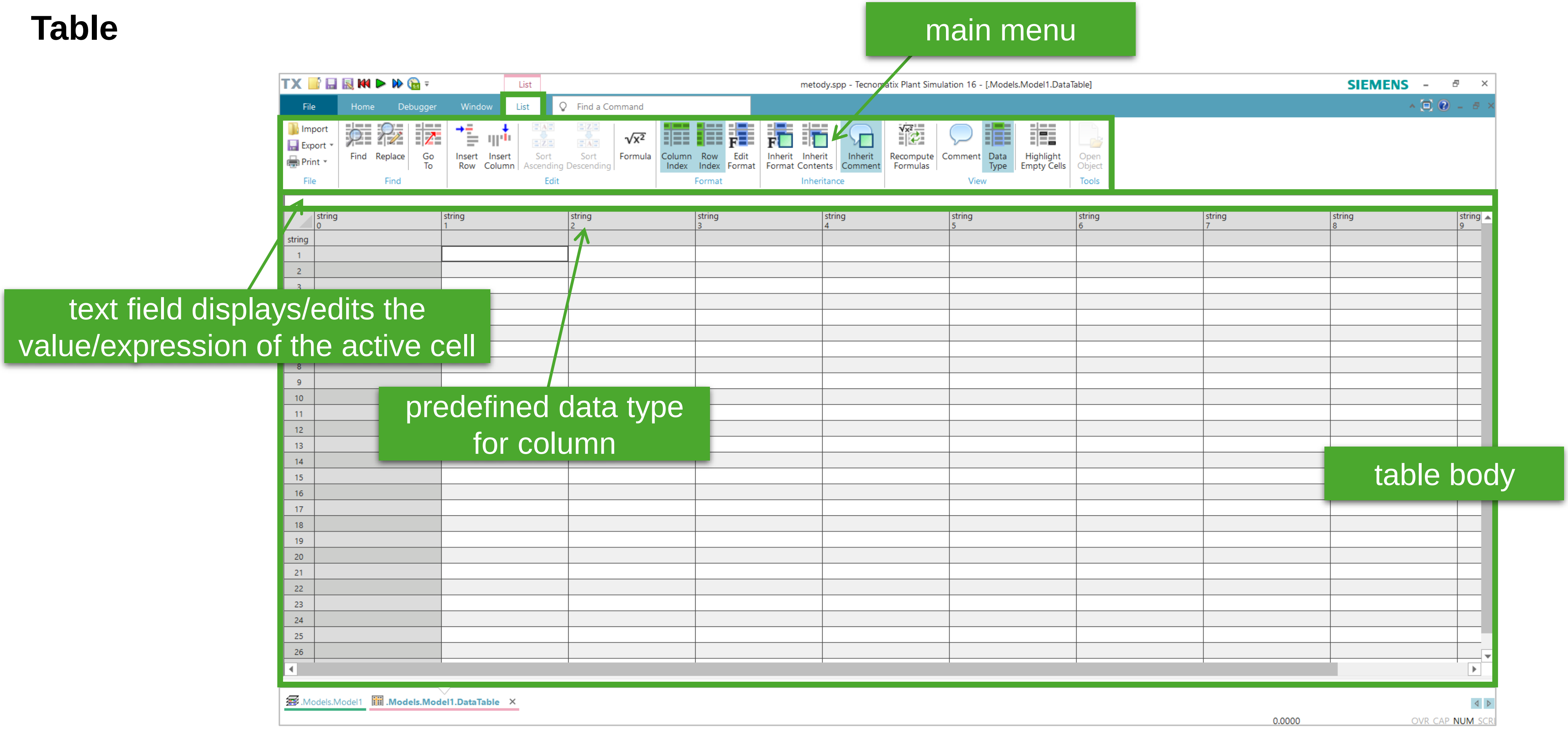
Label:

OK Cancel

Rename (F2)

Programming in SimTalk

Table



main menu

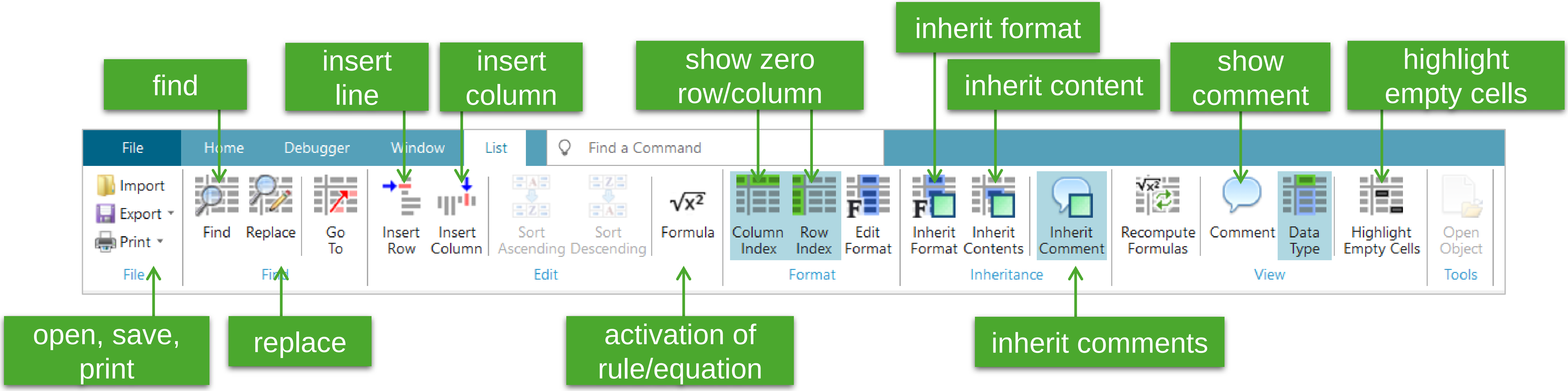
text field displays/edits the value/expression of the active cell

predefined data type for column

table body

Programming in SimTalk

Table – main menu



find

insert line

insert column

show zero row/column

inherit format

inherit content

show comment

highlight empty cells

open, save, print

replace

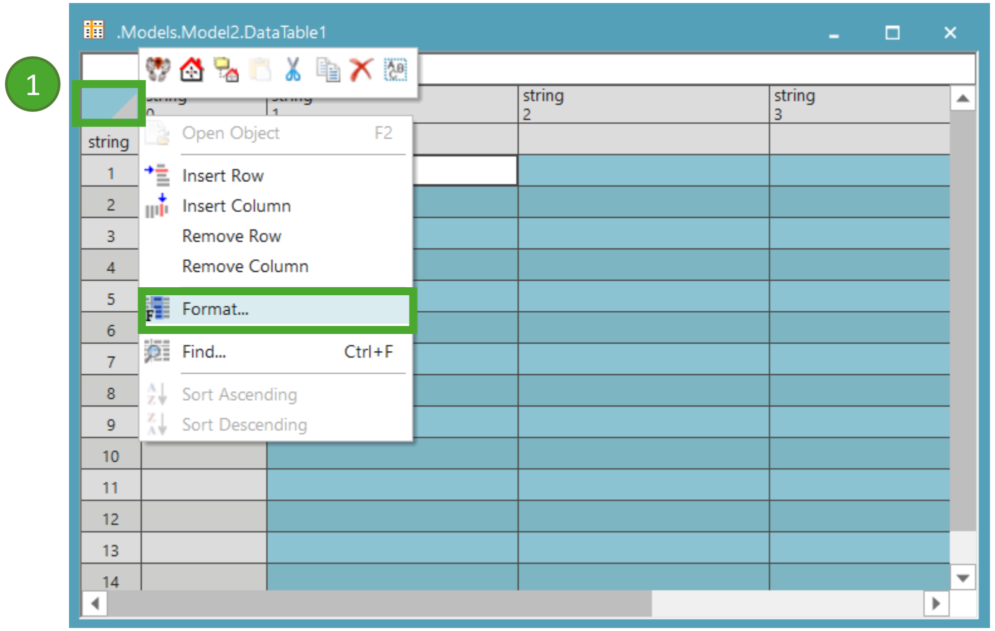
activation of rule/equation

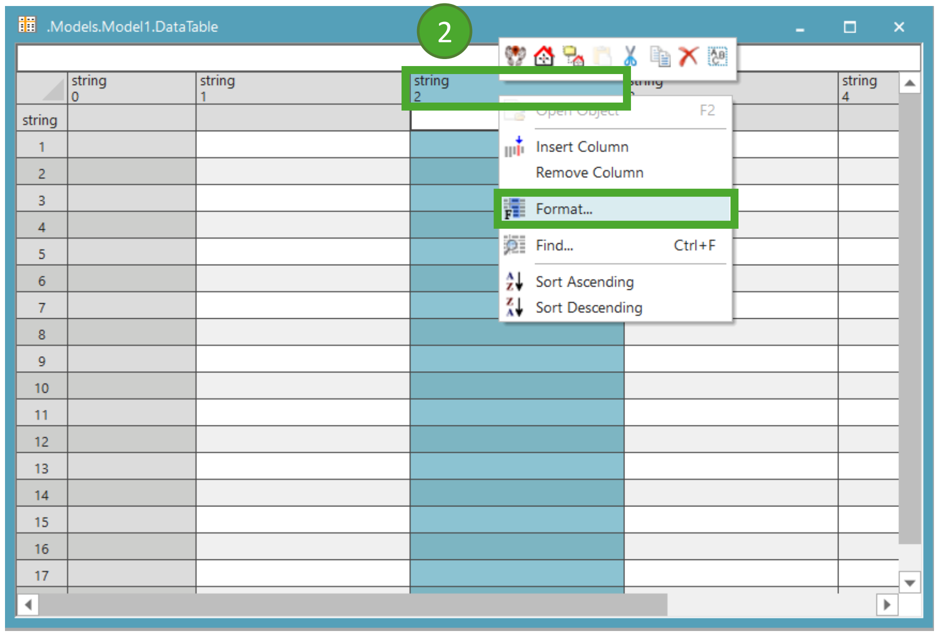
inherit comments

Programming in SimTalk

Table format

- Table format
  - Possibility to format the entire table (1) or rows / columns only (2).



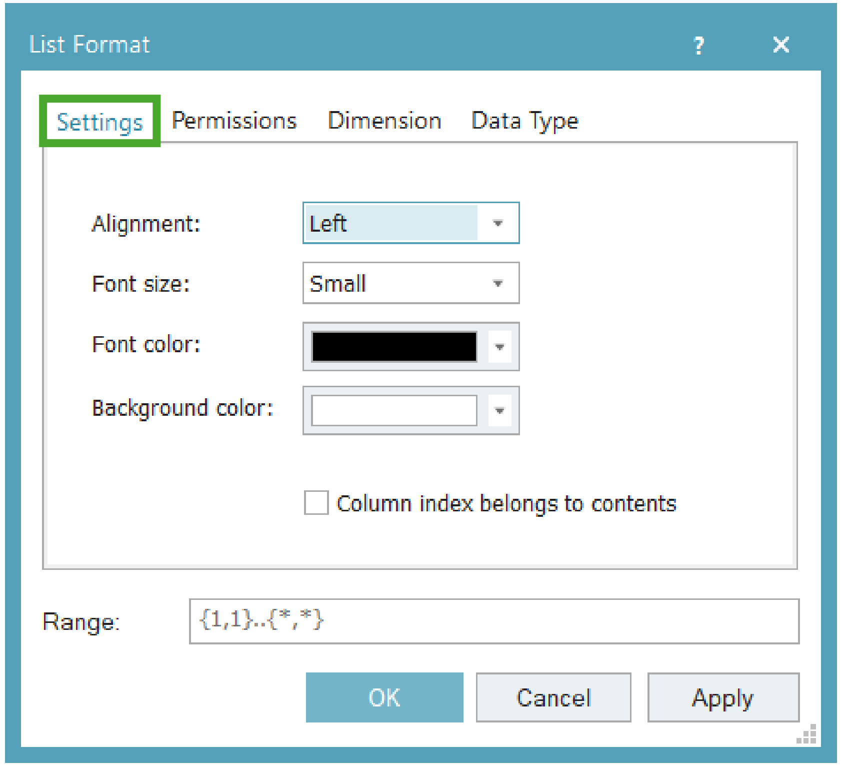


  - After activating the Inheritance function, the entire table is reformatted = DATA LOSS!

Programming in SimTalk

Table format

- Table format
  - Setting of alignment, font size and color, cell color.



Alignment: Left

Font size: Small

Font color: ■

Background color: □

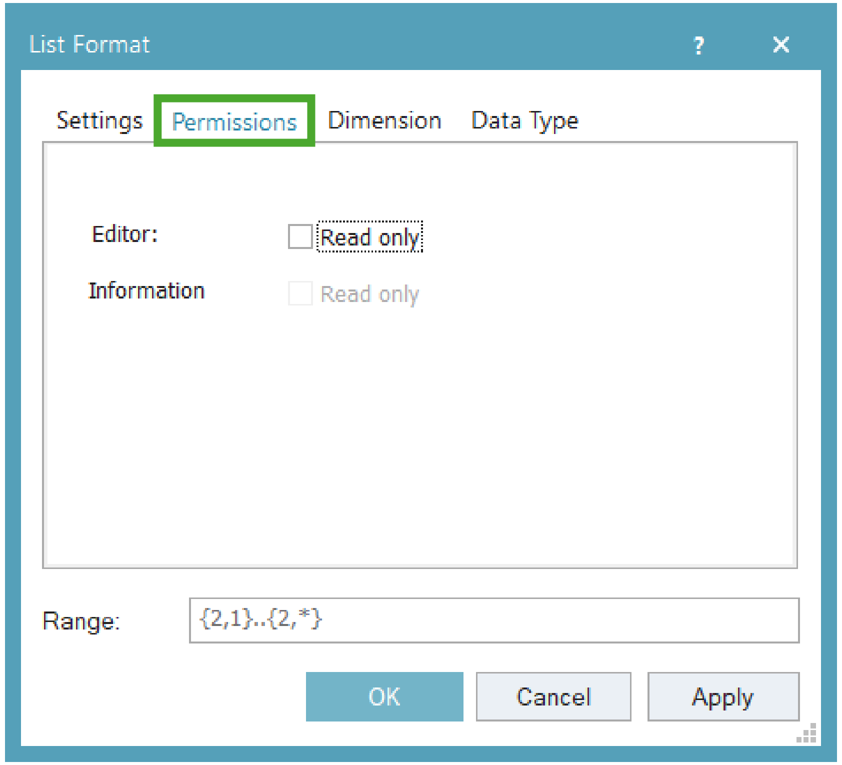
□ Column index belongs to contents

Range: {1,1}..{*,*}

Programming in SimTalk

Table format

- Table format
    - Setting of write and read-only permissions.



Editor: □ Read only

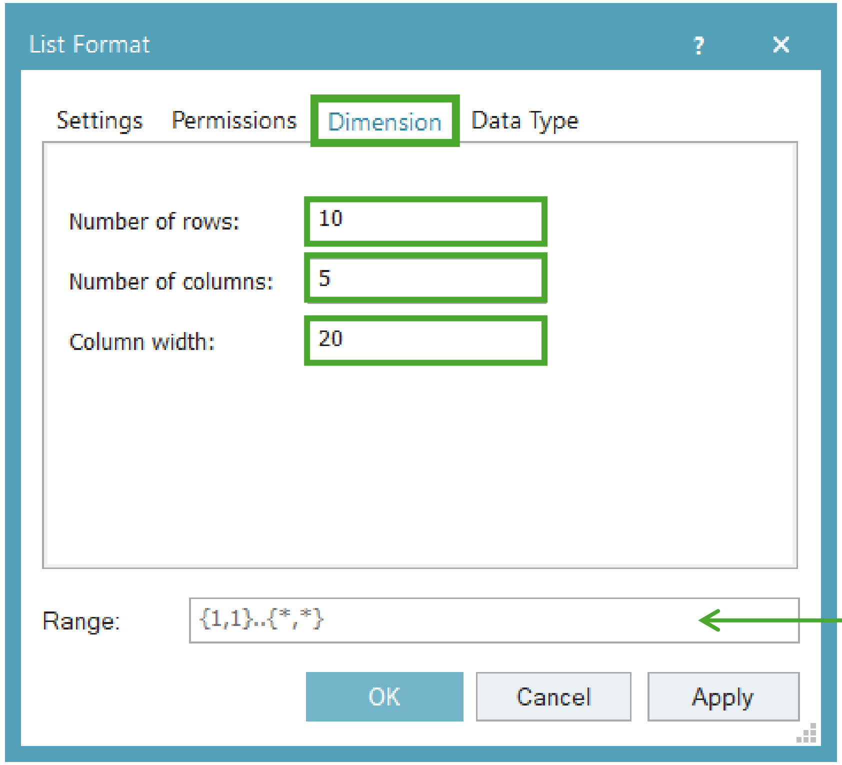
Information □ Read only

Range: {2,1}..{2,*}

Programming in SimTalk

Table format

- Table format
  - Edit number of rows/columns.



Number of rows: 10

Number of columns: 5

Column width: 20

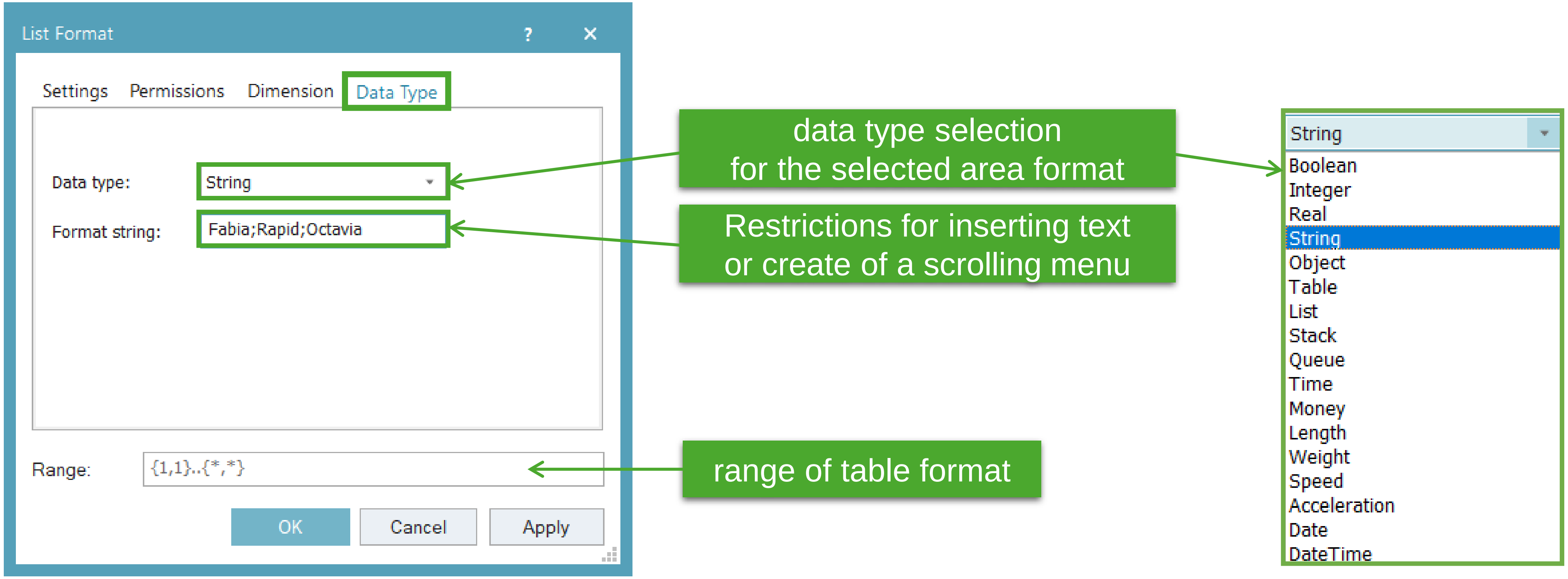
Range: {1,1}..{*,*}

range of table format

Programming in SimTalk

Table format

- Table format
    - Modification of data type.



Data type: String

Format string: Fabia;Rapid;Octavia

Range: {1,1}..{*,*}

data type selection for the selected area format

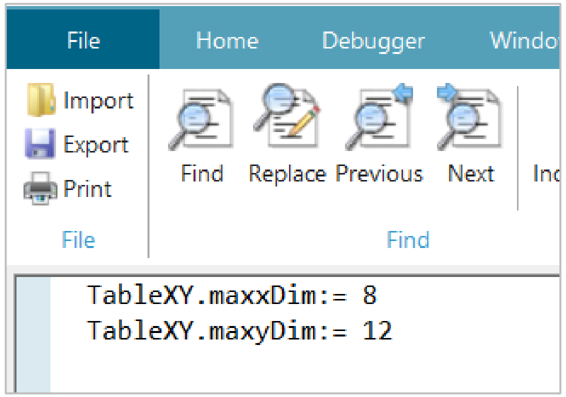
Restrictions for inserting text or create of a scrolling menu

range of table format

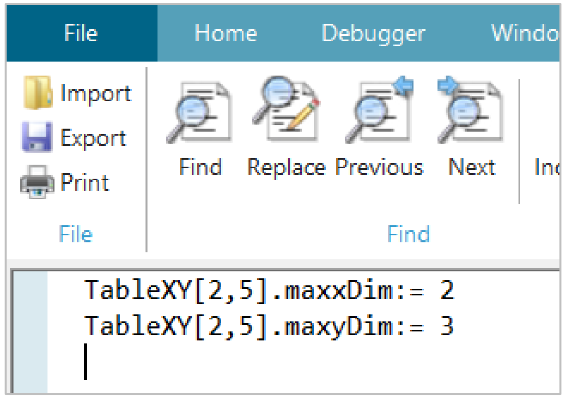
Programming in SimTalk

Table format via method

- Table format
  - Table formatting via method
    - Numbers of columns, rows
      - <table_name>.MaxxDim := 8
      - <table_name>.MaxyDim := 12
  - SUBtable formatting via method
    - Numbers of columns, rows
      - <table_name>[column,row].MaxxDim := 2
      - <table_name>[column,row].MaxyDim := 3



```
TableXY.maxxDim:= 8
TableXY.maxyDim:= 12
```



```
TableXY[2,5].maxxDim:= 2
TableXY[2,5].maxyDim:= 3
```

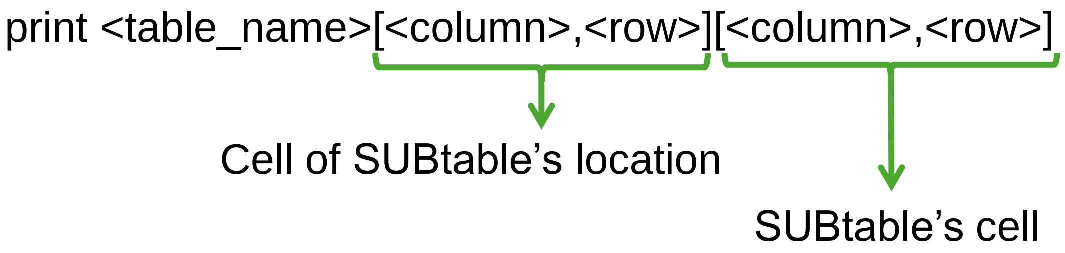
Programming in SimTalk

Table format via method

- Reading from table
    - print <table_name>[<column>,<row>]
- General write to the table
    - <table_name>[<column>,<row>]:= <value>
- Change of object’s attributes from the table
    - <Object>.<attribute> := <table_name>[<column>,<row>]
- Deleting of the contents of the table
    - <table_name>.delete
    - <table_name>.delete({<column>,<row>}…{<column>,<row>})
- When the zero column and row are active and filled, it is possible to enter a description as a path to the cell
    - <table_name>["Object",”Pass_time"]

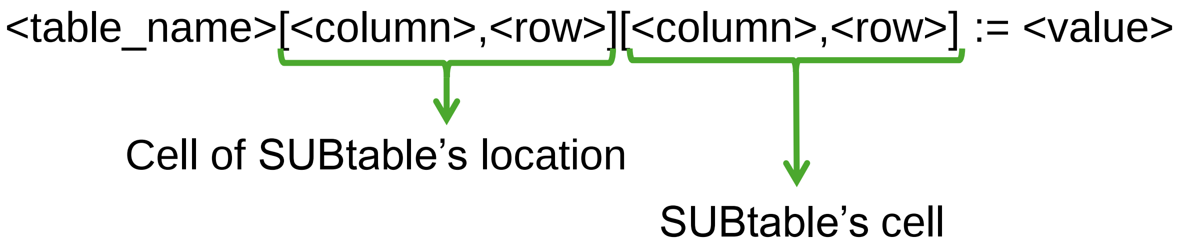
Programming in SimTalk

Table format via method

- Reading from SUBtable
    - 

      print <table_name>[<column>,<row>][<column>,<row>]

      Cell of SUBtable’s location

      SUBtable’s cell
- General input to the SUBtable
    - 

      <table_name>[<column>,<row>][<column>,<row>] := <value>

      Cell of SUBtable’s location

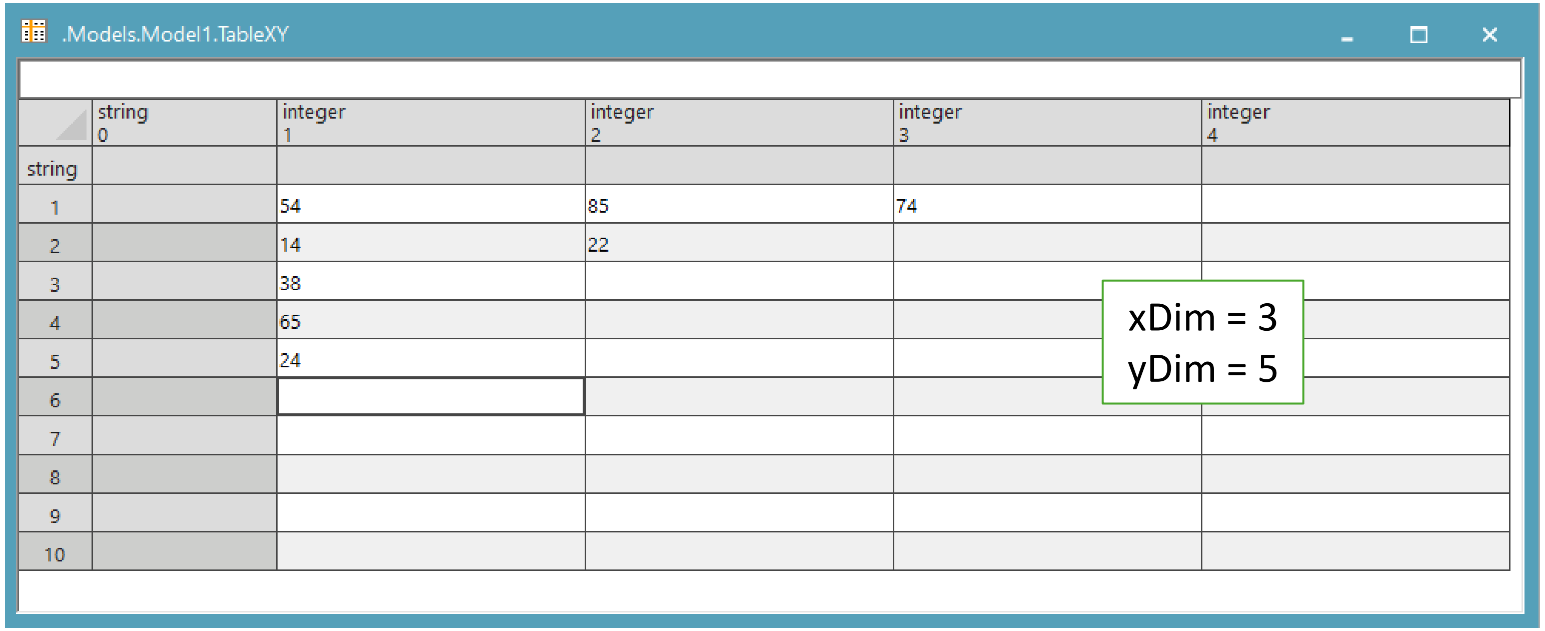
      SUBtable’s cell

Programming in SimTalk

ŠKODA AUTO University

Table

- xDim, yDim
  - It returns the number of the last column, row in the table, which contains a record. The number does not contain zero column and row.
  - Values are of data type integer.



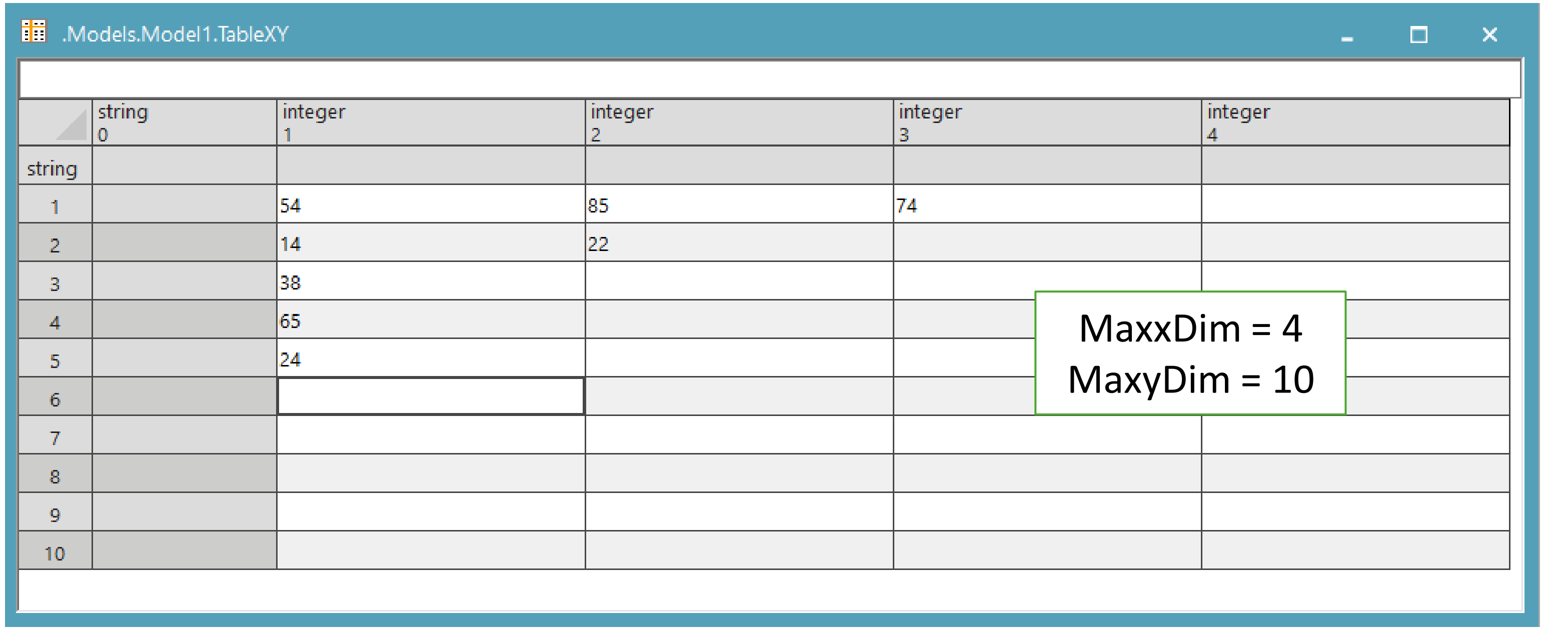
	string 0	integer 1	integer 2	integer 3	integer 4
string					
1		54	85	74	
2		14	22		
3		38			
4		65			
5		24			
6					
7					
8					
9					
10					

xDim = 3
yDim = 5

Programming in SimTalk

Table

- MaxxDim, MaxyDim
  - It returns the absolute number of columns, rows in the table. The number does not contain zero column and row.
  - Values are of data type integer.



.Models.Model1.TableXY

	string 0	integer 1	integer 2	integer 3	integer 4
string					
1		54	85	74	
2		14	22		
3		38			
4		65			
5		24			
6					
7					
8					
9					
10					

MaxxDim = 4
MaxyDim = 10

ŠKODA AUTO University

Thank you for attention

Jan Fábry

Department of Production, Logistics and Quality Management

✉ fabry@savs.cz 🌐 www.janfabry.cz

www.savs.cz