camos.

camos Develop
Developer training

Basics CLS

camos.

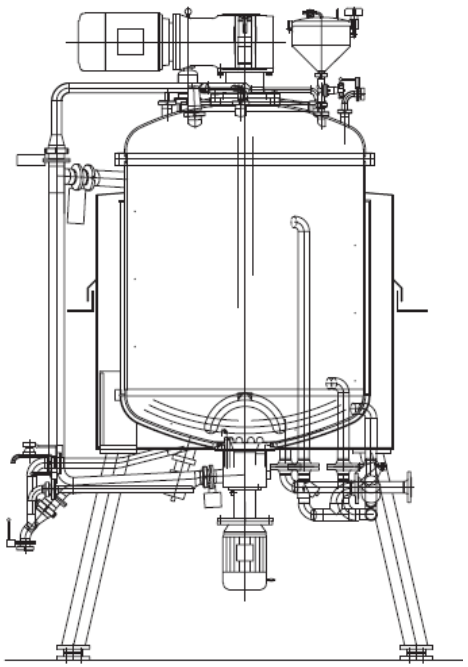
Prerequisites and contents

- **Prerequisites**
  - Car configurator on the state 3. day modeling training
- **Target**
  - Selection of a tyre model by means of certain filter criteria
- **After these exercises you should …**
  - Know the advantages of CLS
  - Be able to define classification tables and fill them with data
  - Know when a classification trigger is firing

Classification

CLS = CLassification System

- **Classification** means the combining of objects to classes
- Each class is identified by certain properties



Manufacturer

Pressure concept

Revolutions

Safety

Shaft

Effective volume

Height

Diameter

Voltage

Transmission

Power

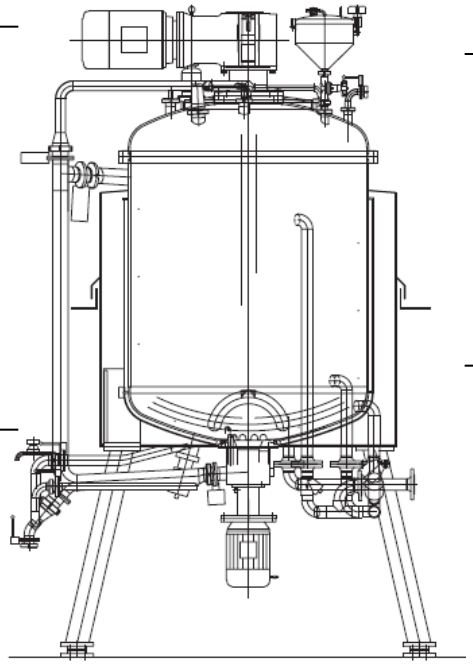
Add-on pieces

Classification

- A classification is the assembling of any amount of relevant properties of the class
- Materials can be determined by means of the property values

Machine
- Type
- Manufacturer
- Pressure
- Safety

Container
- Height
- Diameter
- Effective volume



Engine
- Voltage
- Revolutions
- Power

Gear
- Transmission
- Shaft

Add-on pieces

camos.

Classification in camos Develop

- **A classification consists of two parts**
  - The classification tables
    - contain combinations of criteria values
    - these describe a material
  - The classification trigger defines
    - which values are compared
    - to which features / components the values are assigned
    - when the assignment is running

camos.

Advantages

- Classification data in the development system
- Classification system globally accessible
- Data maintenance also in the application (Quotation)
- Table data is saved as BLOB and loaded into the cache if the data is used
  - less DB-accesses

CLS prerequisites

- **OCL-license**
  - Definition of the CLS-tables and CLS-triggers always possible
  - Special license only for CLS (and OCL) but without rules
    - camos.OCL
  - If CLS-trigger is ruled, an additional rule license is required
    - if license is not present, trigger is “always allowed”

- **When does the trigger fire?**
  - The CLS-trigger is processed if the value of a Wasele (criterion) that is allocated in the CLS-trigger changes
  - Important: Option “OCL- and CLS-criteria are check-relevant” in the KNB-properties

Exercise: Create reference features

- **With the maintenance of a CLS-table**
  - the possible values can be entered freely
  - can be selected from possible values (of a Wasele)
    - to do so, the connection to a feature / component in the knowledge base is necessary

- **Create the following list features in the class Tyres:**
  - ArticleNo[] S
  - Width[] 1
  - Diameter[] 1
  - Type[] S
  - Manufacturer[] S
  - Description[] S
  - UnitPrice[] 1

Exercise: Values of the reference features:

- **Numerical and string values can be created easily in copying them from an Excel file**

	A	B	C	D	E	
1	ArticleNo	Width	Diameter	Type	Manufacturer	Descri
2	R-1-1	155	13	AS	Roadstar (Jupiter)	ALLSE
3	R-1-2	175	14	A	Kingstar Bodyguard (by Hankook)	H714 9
4	R-1-3	185	15	S	Continental	QUATF
5	R-2-1	205	16		Pirelli	Premiu
6	R-2-2		17		FireStone	ECO C
7	R-2-3				MATADOR	HP188
8	R-3-1				Vredestein	K106 9
9	R-3-2				Antyre	WINTE
10	R-3-3					SNOW

- **Insert values from Excel file:**
  - Copy values from column of the Excel file
  - Insert in value range
  - → Values are created automatically

camos.

Exercise: Create CLS-table

- **The structure of the table (columns = criteria) has to be determined before the value combinations are defined**
  - Open classification system (icon or menu Administrations)
  - Create new table “Tyres”
  - In the structure view the criteria
    - can be created manually
    - or via D&D of the features / components from the structure tree

camos.

Exercise: Create CLS-table

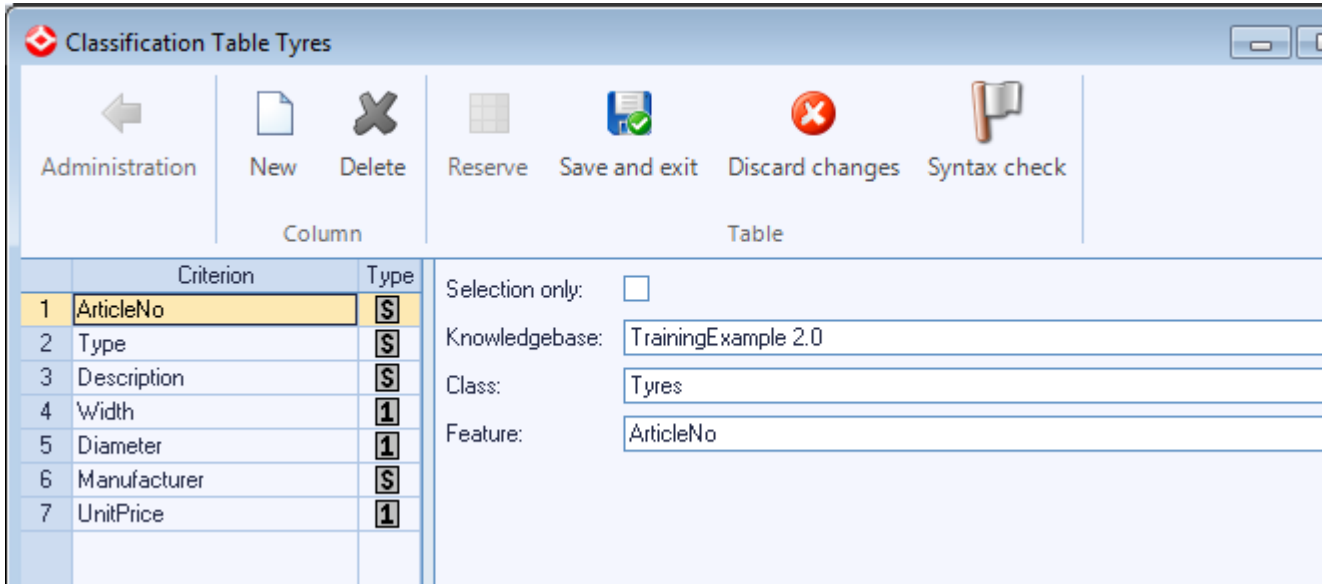
- **Create criteria for all just created features**

- **Important:**
  - Set type correctly
  - Deposit reference for KNB, class, feature
  - -> is set automatically with the creating via D&D

- **Option “Selection only”**
  - Only the values of the Wasele can be assigned in the table
  - No free entry possible

Exercise: Create CLS-table

- **Structure definition of the CLS-table Tyres:**



- **Close the structure view via „Save and exit“**

camos.

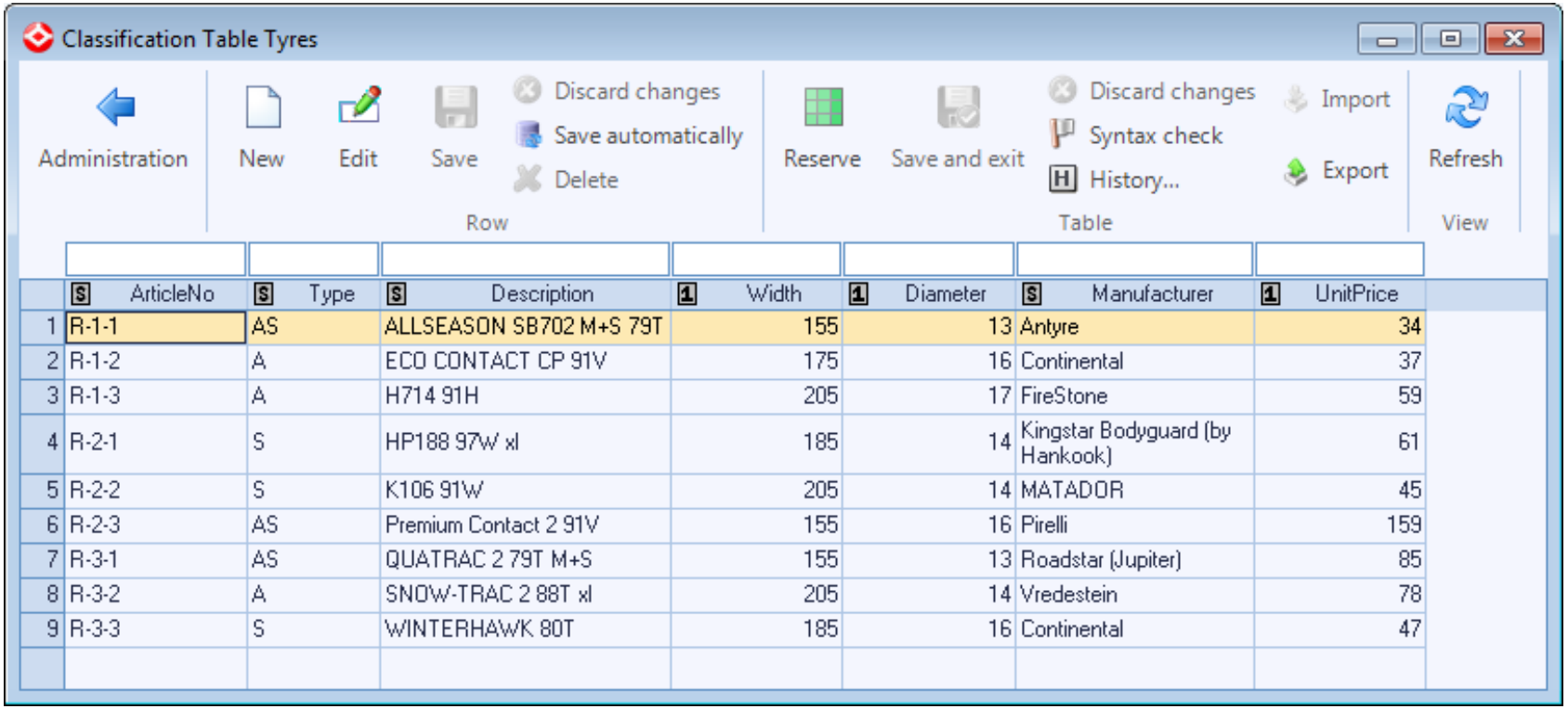
Exercise: Data acquisition in CLS table

- **To acquire the data:**
  - Double-click table
  - Icon or context menu „Edit“

- **Editing:**
  - The table can be locked completely
    - then no other user can make changes at the same time
  - Data records can be edited individually
    - then the remaining data records can be changed by other users at the same time

Exercise: Data acquisition in CLS table

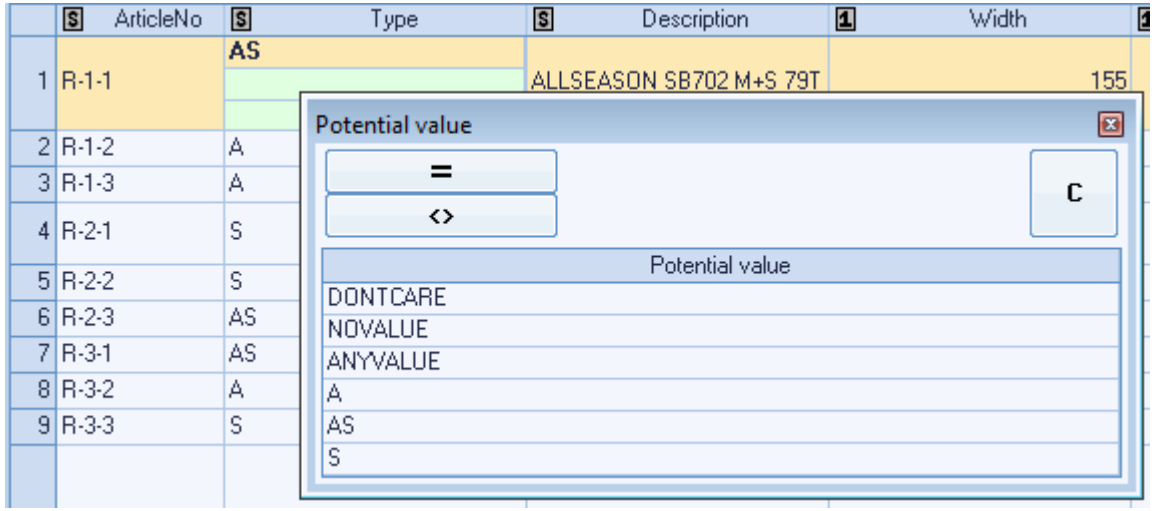
- Create the data by means of the following figure:



	ArticleNo	Type	Description	Width	Diameter	Manufacturer	UnitPrice
1	R-1-1	AS	ALLSEASON SB702 M+S 79T	155	13	Antyre	34
2	R-1-2	A	ECO CONTACT CP 91V	175	16	Continental	37
3	R-1-3	A	H714 91H	205	17	FireStone	59
4	R-2-1	S	HP188 97W xl	185	14	Kingstar Bodyguard (by Hankook)	61
5	R-2-2	S	K106 91W	205	14	MATADOR	45
6	R-2-3	AS	Premium Contact 2 91V	155	16	Pirelli	159
7	R-3-1	AS	QUATRAC 2 79T M+S	155	13	Roadstar (Jupiter)	85
8	R-3-2	A	SNOW-TRAC 2 88T xl	205	14	Vredestein	78
9	R-3-3	S	WINTERHAWK 80T	185	16	Continental	47

Editing CLS-tables

- **Value list shows possible values**
  - if the reference to a Wasele of the knowledge base was deposited in the structure



- **Free entry**
  - as far as allowed

- **Special values**
  - NOVALUE
  - ANYVALUE
  - DONTCARE = NOVALUE or ANYVALUE

camos.

Target definition

- **Form on which you can filter for the properties of the tyres**
  - According to width, diameter, unit price
- **The more the selection is limited, the less tyre models can be selected**
  - Filtered selection is displayed in table
- **Selection of a tyre**
  - Description is displayed in the result

camos.

Exercise: Form TyreFilter

- **Copy features (class Tyres)**
  - Copy the features Width, Diameter and UnitPrice
  - Insert as WidthSelection, DiameterSelection and UnitPriceSelection
  - Remove property List
- **Create form “TyreFilter” (class Tyres)**
- **Create 3 configboxes**
  - Cause variables: WidthSelection, DiameterSelection and UnitPriceSelection
  - Representation Compact

Exercise: Form TyreFilter

- **Create table**
  - One label column per list feature
  - Define cause variables
  - Set heading

TyreFilter

Width: WidthSelection

Diameter: DiameterSelection

Unit price: UnitPriceSelection

Article No	Description	Width	Diameter	Manufacturer	Type	Unit price
Ascii_1	Ascii_1	Ascii_1	Ascii_1	Ascii_1	Ascii_1	Ascii_1
Ascii_2	Ascii_2	Ascii_2	Ascii_2	Ascii_2	Ascii_2	Ascii_2
Ascii_3	Ascii_3	Ascii_3	Ascii_3	Ascii_3	Ascii_3	Ascii_3
Ascii_4	Ascii_4	Ascii_4	Ascii_4	Ascii_4	Ascii_4	Ascii_4
Ascii_5	Ascii_5	Ascii_5	Ascii_5	Ascii_5	Ascii_5	Ascii_5

camos.

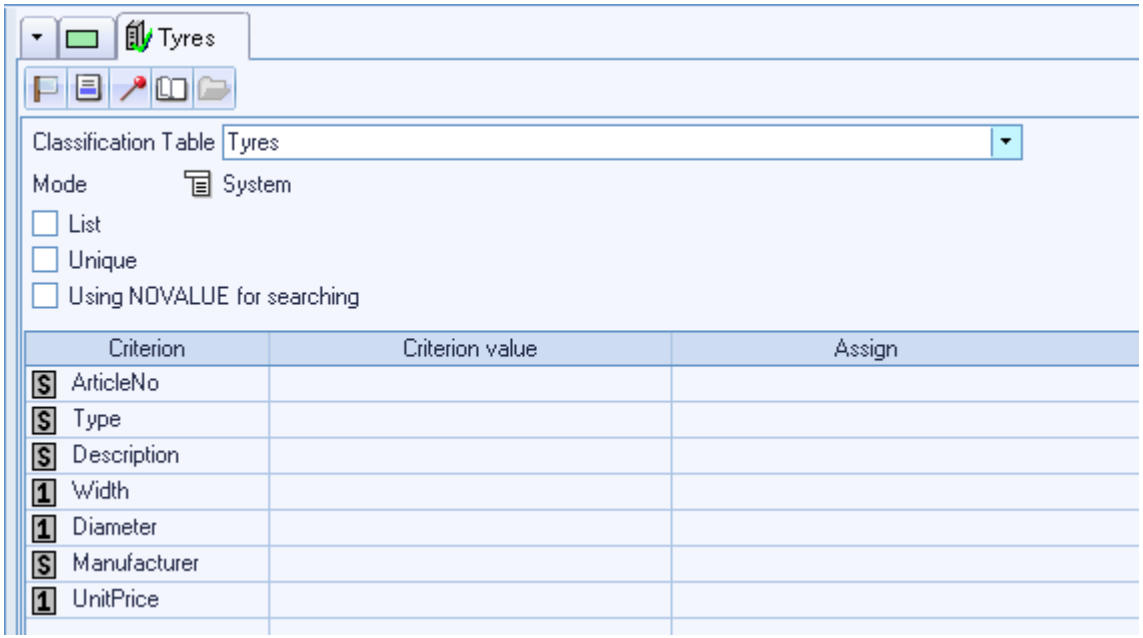
Exercise: Create classification trigger

- **A classification trigger is created in order to trigger an action during runtime**
  - this trigger points to a CLS-table
  - defines which values are used as comparison values (= criteria)
  - and to which Wasele the determined values are assigned

- **General:**
  - A class can contain any amount of CLS-triggers
  - The CLS-trigger can be ruled additionally

Exercise: Create classification trigger

- **Create a classification trigger in the structure tree**
  - this trigger points to the classification table “Tyres”
  - criteria of the table displayed in the CLS-trigger



camos.

Classification trigger

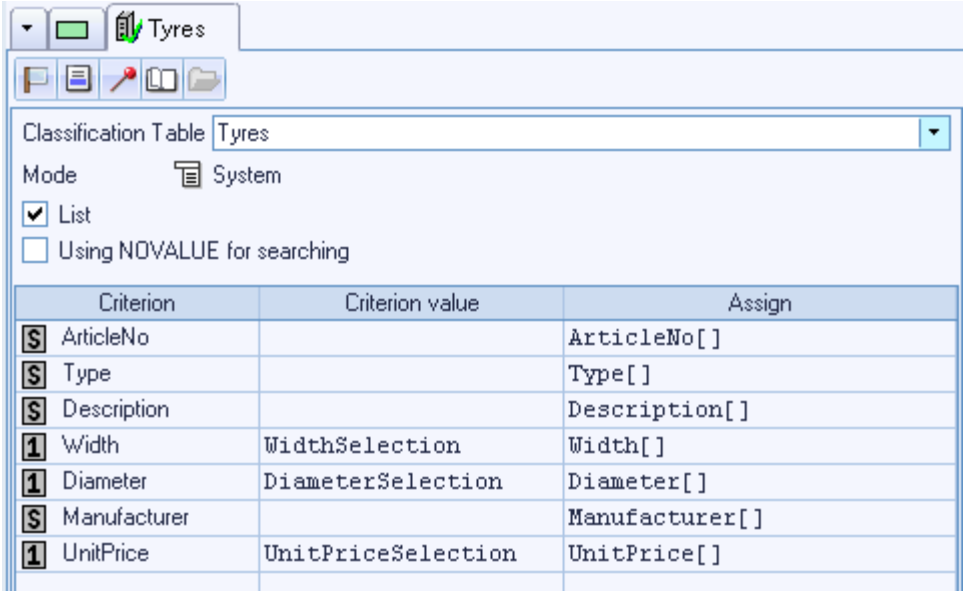
- **When does the classification trigger fire?**
  - If one of the filter values (= criterion values) changes
  - Criterion values have to be check-relevant if the option with the same name is set in the KNB-properties

- **Which action should be triggered by the classification trigger?**
  - Determining the possible tyre models by means of the filter criteria
  - Assigning the values from the CLS-table to the list features

Exercise: CLS-trigger definition

- **Column Criterion value**
  - Defines when the trigger should be fired

- **Column Assign**
  - Defines the action = Assigning the values to the Wasele



Criterion	Criterion value	Assign
ArticleNo		ArticleNo[]
Type		Type[]
Description		Description[]
Width	WidthSelection	Width[]
Diameter	DiameterSelection	Diameter[]
Manufacturer		Manufacturer[]
UnitPrice	UnitPriceSelection	UnitPrice[]

camos.

Options of the CLS-trigger

- **List**
  - The trigger returns several values
  - The assignment is carried out to a list Wasele
- **Unique**
  - Trigger returns one value
  - If more than one value is found, NOVALUE is assigned
- **Using NOVALUE for searching**
  - 1: The value NOVALUE is also searched
  - 0: Criteria with value NOVALUE are ignored with the search

camos.

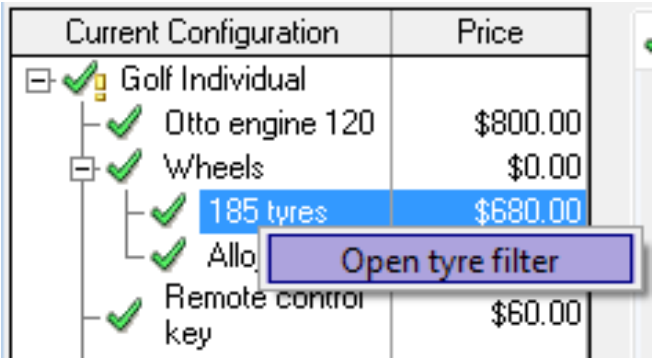
Prerequisite

- **Since the data of the CLS-tables is read from the database, the DSN has to be known during runtime**
  - In the new-method in Tyres:
    `SystemSet('DSNCLS', <DSN-name>);`
  - Here the DSN of the Develop database is used
  - Any DB can be used as far as it contains the CLS-tables (e.g. Quotation DB)
- **SystemSet call in new-method in the class Tyres**

Exercise: Call form

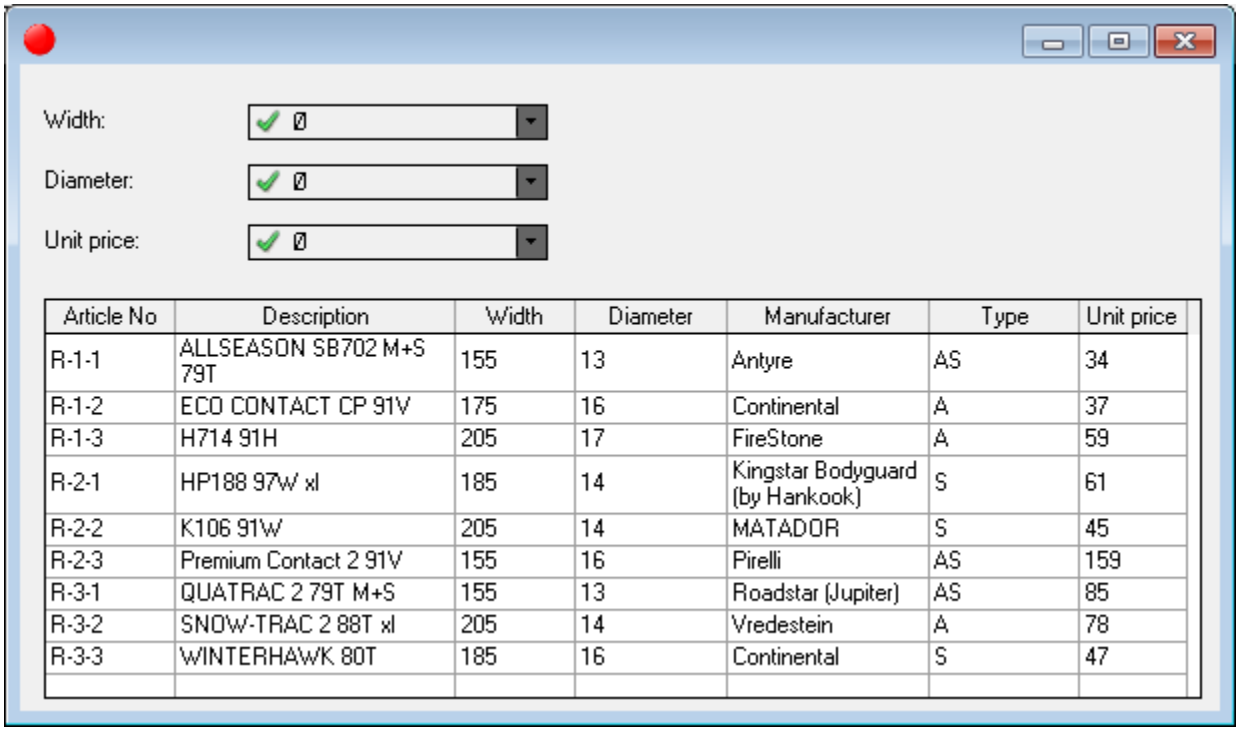
- **Create menu „Standard“ in Tyres**
  - Menu item „Open tyre filter“
  - Opens the form „TyreFilter“

- **Note: If a menu with the name „Standard“ exists in an object, this menu is displayed automatically in the component tree as context menu of this object**



Current state

- **Test the call of the form**
  - Test different values for width, diameter and unit price
  - -> The selection only displays the matching records



Article No	Description	Width	Diameter	Manufacturer	Type	Unit price
R-1-1	ALLSEASON SB702 M+S 79T	155	13	Antyre	AS	34
R-1-2	ECO CONTACT CP 91V	175	16	Continental	A	37
R-1-3	H714 91H	205	17	FireStone	A	59
R-2-1	HP188 97W xl	185	14	Kingstar Bodyguard (by Hankook)	S	61
R-2-2	K106 91W	205	14	MATADOR	S	45
R-2-3	Premium Contact 2 91V	155	16	Pirelli	AS	159
R-3-1	QUATRAC 2 79T M+S	155	13	Roadstar (Jupiter)	AS	85
R-3-2	SNOW-TRAC 2 88T xl	205	14	Vredestein	A	78
R-3-3	WINTERHAWK 80T	185	16	Continental	S	47

camos.

Exercise: Initially value the width of the selected tyres

- **Target:**
  - The width of the selected tyre should already be set in the form TyreFilter
  - To do so, the feature WidthSelection is initially valued for each tyre class

- **Important:**
  - Enable „Reinitialize when object changes“ on the feature WidthSelection in Tyres

camos.

Exercise: Apply selection

- **The naming of the selected tyre model should appear in the component tree and the offer**
  - To do so, the selected line should be known
  - Create numerical feature selLine
  - Deposit on table in the field „Selected“

  - Create pushbutton OK and Cancel on form
  - Selection trigger Cancel:
    - `WinClose(WinGetHandle());`

Exercise: Apply selection

- **With the selection the description is set as naming and the price is adapted**
  - Selection trigger OK:

```
IF selLine THEN
ObjSetNaming(°Tyres° & ': ' & Description[selLine]);
Price := 4 * UnitPrice[selLine];
  WinClose(WinGetHandle());
ELSE
WinMessage('Info', 'No tyres selected');
ENDIF;
```

  - Set the init value of price for all tyres to e.g. 100
    - -> Basic price
    - -> End price only after selection of the model

camos.

Own method CLSTrigger()

- **By default the mode „System“ is set in the CLS-trigger**

- **In order to define further actions before / after / instead of the assignment, the mode can be set to „Own method CLSTrigger“**
  - Through this the system method CLSTrigger() is called with the firing of the trigger
  - Here several properties of the trigger are available in form of parameters
  - Here the function CLSSelect can be called

Exercise: Own method CLSTrigger()

- **Set the mode in the CLS-trigger to „Own method CLSTrigger“**
- **Create the method „CLSTrigger“ via the context menu „New system method“ in the Wasele list**
  - The method contains the following parameters

CLSTrigger()

	Parameter		Class	Call by
1	**TriggerName**	S		**Value**
2	TableName	S		Value
3	IsList	1		Value
4	IsUnique	1		Value
5	NovalueSearch	1		Value
6	CriterionName[]	S		Value
7	CriterionRange[]	S		Value

Exercise: Own method CLSTrigger()

- **The parameters contain the name of the trigger, the CLS-table that is used in the trigger etc. and can be transferred to the function CLSSelect**
- **The function CLSSelect returns a result quantity for a defined return criterion**
  - That´s why the function has to be called for all lists to be filled

```
CLSSelect(ArticleNo[], TableName, CriterionName[], CriterionRange[],
   'ArticleNo', NovalueSearch);
CLSSelect(Description[], TableName, CriterionName[], CriterionRange[],
   'Description', NovalueSearch);
CLSSelect(Width[], TableName, CriterionName[], CriterionRange[],
   'Width', NovalueSearch);
CLSSelect(Diameter[], TableName, CriterionName[], CriterionRange[],
   'Diameter', NovalueSearch);
CLSSelect(Manufacturer[], TableName, CriterionName[], CriterionRange[],
   'Manufacturer', NovalueSearch);
CLSSelect(UnitPrice[], TableName, CriterionName[], CriterionRange[],
   'UnitPrice', NovalueSearch);
CLSSelect(Type[], TableName, CriterionName[], CriterionRange[], 'Type',
   NovalueSearch);
```

camos.

Own method CLSTrigger()

- **Optionally further actions can be carried out additionally in the method**
  - E.g. SQL-statement on further database tables to determine a validity date