camos.

camos Develop
Developer training

Procedural features

camos.

Prerequisites

- **Carconfigurator on the state 3. day modeler training**

- **Contents:**
  - procedural features
  - Read procedure
    - Financing: Calculate repayment installment
  - Write procedure
    - Financing: Enter repayment installment

camos.

Training targets

- **After these exercises you should…**
  - Name the advantage of procedural features
  - Use the read procedure
  - Use the write procedure

camos.

Procedural features

- **Procedural feature**
  - A procedural feature calculates its value independently via the read procedure that is deposited on the feature
  - The read procedure is processed automatically if a persistent value in the object tree changes

- **Advantages**
  - The feature always has a current value
  - The calculation does not have to be triggered manually
  - The calculated value is cached

- **Exceptional features**
  - Procedural features cannot have a initialization
  - Internal and external access has to be considered

camos.

Exercise: Read procedure

- **Target**
  - An installment payment in which the number of installments and an optional down payment can be determined freely should be calculated for the end price
- **Problem**
  - When does the repayment installment have to be calculated?
    - When the number of the installments, the amount of the down payment or the end price amount changes
- **Solution**
  - Use of a procedural feature for the repayment installment
    - Facilitates an automatic reaction to changes

Exercise: Read procedure

- **Create the following features in the class „start“**
  - NumberInstallments, numeric, init value: 12
  - DownPayment, currency, init value: 0
  - Repayment, currency, enable „Procedural feature“

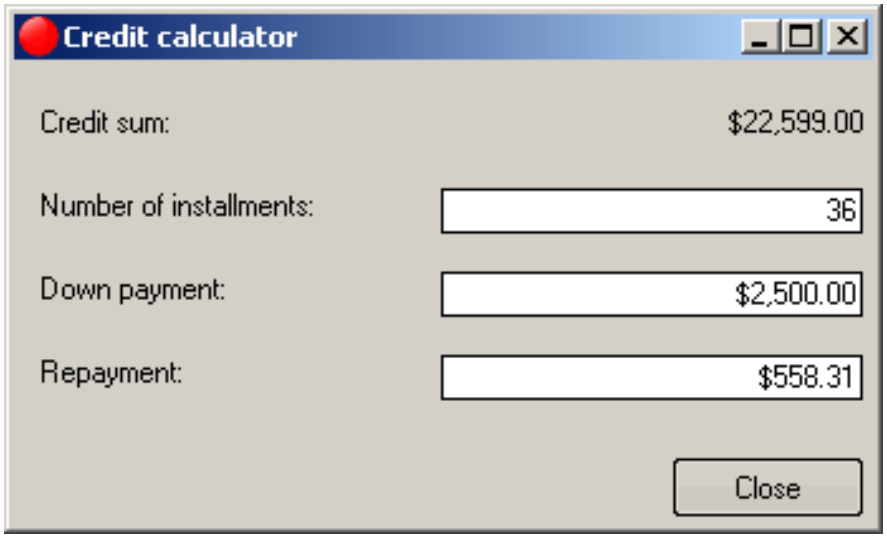
- **„Read procedure“ of the feature „Repayment“**

```
RETURN (_Car.EndPrice - DownPayment) / NumberInstallments;
```

- **Create form handle**
  - Create a new numerical feature „finHandle“

Exercise: Read procedure

- **Create the form „Financing“ and create:**
  - Label static with text „Credit sum“ and postfix „:“
  - Dynamic label with cause variable „_Car.EndPrice“
  - Editline for entering the number of installments
  - Currency element for entering the down payment
  - Currency element for displaying the repayment amount
  - Three (naming) labels with postfix „:“



Exercise: Read procedure

- **Closing the form „Financing“**
  - Create a pushbutton „Close“
    ```
    WinClose(finHandle);
    ```
  - In the Close-trigger of the form you define:
    ```
    finHandle := NOVALUE;
    ```
- **Extend the menu „Administration“ by a menu trigger**
  - Naming: „Financing“
  - Enabled: _Car <> NOVALUE
    ```
    IF not finHandle THEN
      finHandle := WinOpen('Financing');
    ENDIF;
    ```

Exercise: Read procedure

- **Test the credit calculation**
  - Start the application, configure a model and click on „Financing“
  - The repayment installment is calculated immediately due to the initially valued runtime and the down payment amount – without an extra call of the calculation
  - Enter values for „Number of installments“ and „Down payment“ and check the correct recalculation of the repayment amount
  - Check if the display of credit sum and repayment is current after changes on the configuration were made

camos.

Exercise: Write procedure

- **Target**
  - Enter repayment installment whereupon the resulting runtime of the installment payment is calculated
- **Problem**
  - Generally procedure values cannot be allocated with a value, because they calculate their value independently
- **Solution**
  - Set access (internal and external) to „Full“
  - Register „Write procedure“ is enabled
    - The internal parameter „Value“ contains the assigned value

Exercise: Write procedure

- **Set internal and external access on „Repayment“ to „Full“**

- **Deposit the following code as „Write procedure“**

```
Runtime := Currency2Num(_Car.EndPrice - DownPayment, 1) /
Currency2Num(Value, 1);
NumberInstallments := rdp(Runtime);
```

  - Create numerical variable „Runtime“

- **Important**
  - Currency values cannot be divided. Therefore they have to be converted to numerical values before this operation is carried out

    -> Currency2Num(Currency value, Country code)

camos.

Exercise: Write procedure

- **Test the write-procedure**
  - Enter any repayment amount and exit the entry field
    - The number of required installments is calculated
    - Then the feature „Repayment“ however has a different value than previouly entered
  - Explanation
    - The read-procedure is executed automatically after the execution of the write-procedure, because a value in the object tree has changed
    - Since the result of the read-procedure was rounded (integer installment number), the monthly repayment has to be recalculated