



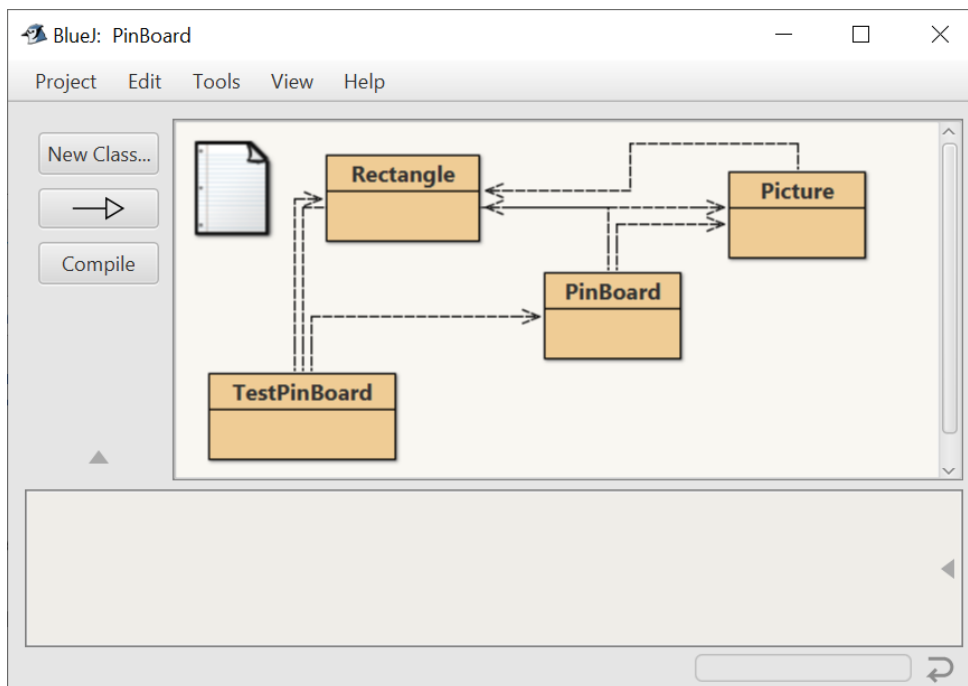
Lab for “Java Programming”

Class: 1EHIF – 1 December 2020

You learn ...

- Implement associations
- Use class Math
- Create proper test cases



Implement the classes **Rectangle**, **PinBoard**, **Picture** and **TestPinBoard** as described below:



The class **Rectangle** has

- the two floating point attributes *length* and *width* (in *cm*) with corresponding get- methods (no set methods – which makes the rectangle immutable)
- a constructor that takes 2 parameters to initialize the attributes. It turns negative values to positive ones (use `Math.abs(..)`) and 0.0 to 1.0. In addition it makes sure that the bigger value is stored in *length* and the shorter one is stored in *width*
- The method `area()` returns the area of the rectangle – use for the calculation

The class **Picture** has

- the attributes (with set- and get methods)
 - *theme* - describes what can be seen in the picture (e.g. house, forest, landscape ...) – must not be null or empty (default is “art”)
 - *size* - a **Rectangle** containing the size of the picture – must not be null (choose suitable values for a default rectangle)
 - *landscape* - whether the picture is landscape (*Querformat* ) or portrait (*Hochformat* ) – default is landscape.

- a constructor that takes the theme and the size as parameter
- The method *display()* ... displays the data of the picture (including, length, width and area of its size!) in proper sentences on the screen

The **PinBoard** has

- the attributes
 - *pic1*, *pic2* and *pic3* - three pictures
 - *size* - a **Rectangle** containing the size of the Pinboard - must not be null (choose suitable values for a default rectangle)
- a constructor that takes only the size as parameter
- set-and get method for size
- The method *hangUpPic (Picture p)* ... puts a picture on the pin board. If 3 pictures are already on the pin board, no further picture can be hung up. The return value indicates whether the action was successful (true) or not (false).
Bonus for professionals: the picture can only be hung up if it additionally does not hide an other picture !
- The method *removePic (int number)* ... removes the given picture according to the number (e.g.: 1... removes pic1) and returns it (if there is no picture null will be returned).
- The method *removePic ()* ... removes the biggest picture and returns it (or null if no picture is available)
- The method *availableSpace()* returns the available space (=> space without pictures) on the pinboard
- The method *display()*... displays the information of the **PinBoard** including its pictures

Implement the class **TestPinBoard** with the methods - make sure that displayed text is easy to understand and makes evident what you test and whether the test succeeded or failed:

- *testRectangle()* where you:
 - create a rectangle with negative values and check if they became positive
 - create a rectangle where width is longer than length and check whether the values were corrected
 - create a rectangle with a width that is 0 or a length that is 0 and check if the value will be corrected
 - create a correct rectangle and check whether *area()* returns the correct size.
- *testBoard()* where you:
 - instantiate four pictures and display them
 - instantiate a pin board
 - hang up the four pictures and display the return value
 - display the pinboard and its available space
 - remove the biggest picture and display the pin board
 - remove another picture and test the return value

When issuing your test, make sure that the text is easy to understand and clear for the user.