Lab 1: Create a model

Task 1: *Graphics.*

- Write a *Hello World* program in the graphical mode.
- Define the package *drawingTool*
- Define a class *TestDrawingTool*
- Define a class *DrawingArea*
- Open *TestDrawingTool.java* and *DrawingArea.java* provided by Emil.
- Don't drag the java files into Eclipse!
- Copy their contents into the source code of the files generated by Eclipse.
- Change the graphical objects and observe what happens. Use Run/Run.
- Find out how to draw a polygon.

Task 2: *Describe your animal, which will be drawn by your Java program.*

- Your description contains:
  - Which kinds of parts (at least 10) are involved? (Think of parts and wholes as well as details, decorations, and variations.)
    (lecture example: Front, Roof, Window, Door)
  - How do they relate?
    (lecture example: Front contains Window)
  - Provide a UML class diagram of your domain
    (drawn by pencil and paper; no tool).
- The implementation of the provided applilcation classes:
  - Use *drawingTool* from task 1.
  - Use the main program of Task 1 (use the code in *TestDrawingTool.java* and *DrawingArea.java*).
  - In the method *paintComponent* of the class *DrawingArea* you construct one object instance of exactly one of your classes, namely the main-class of your animal (e.g. called Cat). This is the entry point of your domain classes.

- The implementation of your own animal classes
  - For each of your identified objects, provide a class (by New/Class separate files are generated for each class with the names of the classes). according to your UML diagram
  - Each class should provide only the relevant properties, a constructor, and a drawAt(int left, int bottom)-method.
    according to your UML diagram
  - In your Cat class you provide a method drawAt(int left, int bottom):
    - from here, you should call drawAt(int left, int bottom)-methods of parts. That is, each class has also its own drawAt(int left, int bottom)-method.
    - draw the parts of the according component (e.g. in class Leg(int left, int bottom) you only would draw exactly one leg;
    - you might additionally call drawAt methods of parts of the leg (e.g. you could conceive the foot being part of the leg according to your UML diagram)
      (hint: you would have four instances of class Leg, for each of the four legs of the cat)
    - Does the result meet your expectations?
    - Where should components of objects be combined?
  - Clean your code, make it perfect
    - look at the last page of this exercise sheet for all code conventions
    - don't use comments, your code needs to look simple, having a lot of short and clean classes, superseding any comments

Software Quality: Code Conventions

a) Identifiers are in English.

b) Identifiers are meaningful, but not too long.

c) Variable identifiers begin with a small letter. Multiple words composed as CamelCase.

d) Identifiers for classes and interfaces begin with a capital letter. Multiple words composed as CamelCase.

e) Identifiers for constants consist only of uppercase letters. Multiple words composed by underline.

f) Left curly braces not in a new line. New line after left curly braces.

g) New line after right curly braces.
Exception: keyword else is in the same line.

h) Logical sections within a method have a comment as a heading.

i) Each block level is horizontally tap-indented by one level.

j) There is a blank line between methods.

k) There is a blank line between classes.

l) Classes and interfaces are separated by a blank line of import and package statements.

m) No more than one blank line in a row.

n) Order within a class or an interface:

1. properties (constants and variables)
2. constructors
3. getter and setter for properties, but only if required
4. other methods