

## Serie 2

### Content:

- Shapes - Reverse engineering (Facultatif)
- UML
- Object manipulation
- Object-oriented programming : “super” and “this”

### (26) Shapes – Reverse Engineering (Optional)

Install the evaluation version of [Visual Paradigm](#). With the help of this application do a reverse engineering of the source-code of the *Shapes* application developed earlier this semester. Observe the automatically generated class diagram.

Create a sequence diagram showing what happens when the menu entry *Move All* is selected.

### (5) UML Class and Object diagrams : Countries

Draw class diagrams expressing that countries have borders with other countries.

- One solution should contain one class only, and a reflexive association.
- Another solution should contain two classes. You may omit attributes.

Consider the countries Portugal, Spain, France, Germany, Switzerland, Italy, and Japan. For both variants of your class diagrams, provide an object diagram.

### (6) UML Class diagrams : Undirected graphs

Draw a class diagram that describes undirected graphs.

An undirected graph  $G$  consists of a set of vertices (or nodes)  $V$  and a set of edges  $E$ , hence  $G = (V, E)$ . Your diagram should capture only the structure of the graphs (i.e., the connectivity), and need not to be concerned with geographical details such as location of vertices and lengths of edges. A typical graph is given on Figure 1.

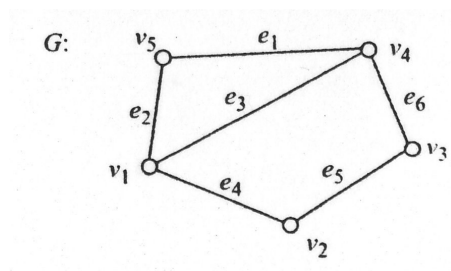


Figure 1 – An example of an undirected graph

### (7) UML Class diagrams : Directed graphs

Draw a class diagram that describes directed graphs. A directed graph is similar to an undirected graph, except that the edges are oriented. An example of a directed graph is shown on Figure 2.

### (8) UML Object and Sequence diagrams : Drawing tool

Let us consider the class diagram (see Figure 3) of a simplified drawing tool in which a sheet is composed of several figures.

Let us assume the existence of the sheet depicted on Figure 4 that contains three figures : a circle, a box and a group. The

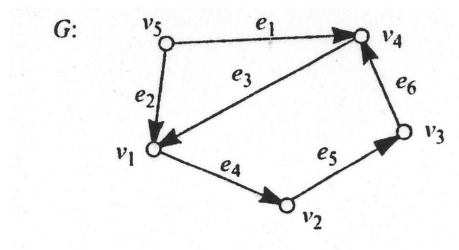


Figure 2 – An example of a directed graph

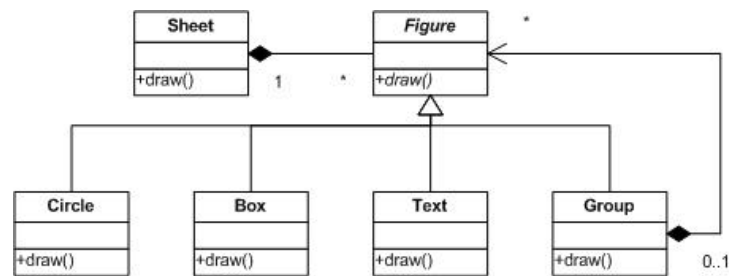


Figure 3 – Drawing tool class diagram

group is composed of 2 figures : another circle and a text.

- First, draw the object diagram of the following sheet ; and then
- Develop a sequence diagram for the invocation of the operation `draw()` on the object sheet.

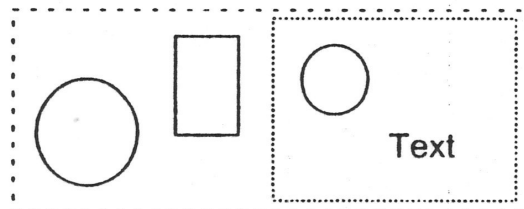


Figure 4 – An example sheet

#### (9) UML Communication diagram : Drawing tool

- Draw a communication diagram for the preceding exercise.
- Discuss the differences between sequence diagrams and communication diagrams.

#### (13) UML Activity Diagram - Chocolate Cake ([1] page 155)

Draw an activity diagram modeling the following chocolate cake recipe.

##### Ingredients :

- 190g cup butter, softened
- 3 eggs
- 250g all-purpose flour
- 100g unsweetened cocoa powder
- 1 teaspoon baking soda
- $\frac{1}{2}$  teaspoon salt

- 500g cups suger
- 2 teaspoons vanilla
- 375ml milk

**Recipe :**

1. Allow butter and eggs to stand at room temperature for 30 minutes. Meanwhile, lightly greas bottoms of one 34x22x4.5-cm baking pan. Line bottom of pans with waxed paper. In a medium bowl stir together flour, cocoa powder, baking soda, baking powder, and salt ; set aside.
2. In a large mixing bowl beat butter with an electric mixer on medium to high speed for 30 seconds. Gradually add sugar, about 50g at time, beating on medium speed until well combined (3 to 4 minutes). Scrape sides of bowl ; continue beating on medium speed for 2 minutes. Add eggs 1 at a time, beating after each addition (about 1 minute in total). Beat in vanilla.
3. Alternately add flour mixture and milk to butter mixture, beating low speed after each addition just until combined. Beat on medium to high speed for 20 seconds more. Spread batter into the prepared pan.
4. Bake in a 180° oven for 35 to 40 minutes, or until a wooden toothpick inserted near centers comes out clean. Cool cake layers in pan for 10 minutes. Remove from pans. Peel off waxed paper. Cool thoroughly on racks. Frost with desired frosting.

**UML Activity Diagram - Cream Cheese Frosting ([I] page 171)**

**Ingredients :**

- 125g package cream cheese, softened
- 40g butter or margarine, softened
- 1 teaspoons vanilla
- 450g sifted powdered sugar

**Recipe :**

1. Beat cream cheese, butter, and vanilla with electric mixer until light and fluffy. Gradually add  $\frac{2}{3}$  of the powdered sugar, beating well. Gradually beat in additional powdered sugar to reach spreading consistency. This frosts tops and sides of one 34x22x4.5-cm cake.

**(10) UML State diagram : Stopwatch**

A stopwatch has two buttons  $B_1$  and  $B_2$ .

Pressing button  $B_1$  starts measuring the elapsed time, the running time is displayed. Pressing button  $B_1$  once more causes the intermediate time to be displayed. However the measurement of the running time continues. When displaying the intermediate time, pressing button  $B_1$  switches to display the running time.

Pressing button  $B_2$  in the initial state of the stop watch has no effect. When displaying the intermediate or running time, pressing button  $B_2$  causes the watch to display the final elapsed time. Pressing  $B_2$  once more resets the stop watch to its idle state.

Draw a state diagram for such a stop watch. Can you identify transitions not mentioned in the preceding description ?

**(11) Object diagram, object manipulation**

Consider the classes `RootClass`, `Class1` et `Class2` (cf. page 3).

1. Draw an object diagram showing the state of the system at the end of the `main` method of the `RootClass` (cf. [II])
2. At position [II], what is the situation of the object that has been created at line [III] of the method `f1` of the class `RootClass` ?

```
1 public class RootClass {  
2  
3     Class1 c1, c2 ;  
4     Class2 s ;  
5  
6     public RootClass () {
```

```

7      s = new Class2("ABC");
8      c1 = new Class1("ABC", s);
9      c2 = f1().f1();           // [11]
10     }
11
12     public static void main(String[] args) {
13         new RootClass();
14     }
15
16     Class1 f1() {
17         Class1 result = new Class1("Cuckoo", new Class2("Cuckoo")); // [111]
18         return (Class1)c1.clone();
19     }
20 }
21
22 class Class1 implements Cloneable {
23
24     int i;
25     String s1;
26     Class2 s2;
27
28     public Class1(String init, Class2 mo) {
29         i = 1;
30         s1 = new String(init); // [1]
31         s2 = mo;
32         s1=s1.concat("DEF");
33         s2.concat("XYZ");
34     }
35
36     public Object clone() {
37         try {
38             return super.clone();
39         } catch (CloneNotSupportedException e) {
40             // This should never happen
41             throw new InternalError(e.toString());
42         }
43
44     public Class1 f1() {
45         s1 = "123";
46         i = i+1;
47         return this;
48     }
49 }
50
51 class Class2 {
52
53     private String contents;
54
55     public Class2(String init) {contents=init;}
56
57     public void concat(String s) {
58         contents=contents.concat(s);
59     }
60
61     public String toString() {
62         return contents;
63     }
64 }

```

Listing 1 – RootClass.java

## (12) Object-oriented programming : “super” and “this”

Consider the classes A, B, C and Start, where Start is the main class of the system :

```

1 public class Start {
2

```

```
3 public static void main(String args[]) {
4     C c = new C();
5     int result = c.zork();
6     System.out.println(result);
7     B b = new B();
8     result = b.zork();
9     System.out.println(result);
10 }
11 }
12
13 class A {
14
15     public int foo() {
16         return 1;
17     }
18
19     public int zork() {
20         return this.bar();
21     }
22
23     public int bar() {
24         return 3;
25     }
26 }
27
28
29 class B extends A {
30
31     public int foo() {
32         return super.zork();
33     }
34 }
35
36
37 class C extends B {
38
39     public int zork() {
40         return this.foo() + 5;
41     }
42
43     public int bar() {
44         return 7;
45     }
46 }
47 }
```

Listing 2 – Source code of classes A, B, C and Start

Which text is displayed if one executes this program ? Explain your answer, explain what happens.

## Références

- [1] Better Homes and Gardens. *New Cook Book*, volume 75th Anniversary Limited Edition (Loose Leaf).
- [2] Jacques Pasquier. Génie logiciel I, 2013. <http://moodle2.unifr.ch/course/view.php?id=1252> (accessed Mar 13, 2013).