

From JAVA to UML

Exercice 1

Question 1 • L'ensemble des classes Java représentent une implémentation d'une modélisation donnée. A partir de ces éléments Java, donner le diagramme de classes correspondant.

```
1 public interface Classe0 {  
    public void methode1(String nom);  
3    public void methode2(Color color) ; }  
  
1 public abstract class Classe1 implements Classe0 {  
  
3    protected int id ;  
    protected static int nombreObjets = 0;  
5    private Classe3 monC3;  
  
7    public void methode1(String s){}  
  
9    abstract public boolean methode3() ;  
  
11    public Classe1(Classe3 c){  
        System.out.println("constructeur de Classe1 avec argument");  
13        monC3 = c;  
        this.id = nombreObjets++;  
15    } }
```

```

1 public class Classe2 extends Classe1 {
    private Classe4 monC4;

3
    public Classe2(Classe3 c3, Classe4 c4) {
        super(c3);
        monC4=c4;
        System.out.println("constructeur de Classe2");
    }

9
    public void methode2(Color color) {}

11
    public boolean methode3() {return true;}

13 }

```

```

1 public class Classe3 {
    private int id ;
    private static int nombreObjets = 0;
    private Collection<Classe1> mesClasses1;

5
    public static int getNombreClasses3() {
        return nombreObjets;
    }

9
    public Classe3() {
        System.out.println("constructeur de Classe3");
        mesClasses1 = new ArrayList<Classe1>();
        id = nombreObjets++;

13 } }

```

```

1 public class Classe4 {
    private int id ;
    private static int nombreObjets = 0;
    private Collection<Classe1> mesC1s;
    private Collection<Classe6> mesC6s;

7
    public Classe4() {
        System.out.println("constructeur de Classe4");
        mesC1s = new ArrayList<Classe1>();
        mesC6s = new ArrayList<Classe6>();
        id = nombreObjets++;
    }

13
    public void createObjet6(Classe5 c5, int i) {
        Classe6 c6 = new Classe6(this, c5, i);
        mesC6s.add(c6);

15 } }

```

```

1 public class Classe5 {
2
3     private int id ;
4     private static int nombreObjets = 0;
5     private Collection<Classe6> mesC6s;
6
7     public Classe5() {
8         System.out.println("constructeur de Classe5");
9         mesC6s = new ArrayList<Classe6>();
10        id = nombreObjets++;
11    }
12
13    public void addObjet6(Classe6 c6) {
14        mesC6s.add(c6);
15    } }

```

```

1 public class Classe6 {
2
3     private int att6;
4     private Classe4 monC4;
5     private Classe5 monC5;
6
7     public Classe6(Classe4 c4, Classe5 c5, int i){
8         System.out.println("constructeur de Classe6, att6 : "+i);
9         monC4 = c4; monC5 = c5; att6=i;
10        c5.addObjet6(this);
11    } }

```

```

1 public class Color {
2
3     int red,green,blue;
4
5     public Color(int r, int g, int b){
6         System.out.println("constructeur de Color");
7         this.red=r;
8         this.green=g;
9         this.blue=b;
10    } }

```

Exercice 2

Question 1 • Récupérer le zip des classes Java **robot.zip** dans moodle et donner le diagramme de classes correspondant.

Exercice 3

Question 1 • Elaborer le diagramme d'objet équivalent au code Java ci-dessous.

```

1  B b = new B()
2  A a = new A2()
3  a.b = true
4  a.c = b
5  a.d = 76
6  A a2 = new A3()
7  a2.d = 87
8  a2.c = new B2()

```

Question 2 • Elaborer le diagramme de classe associé.

Question 3 • Est-il possible d'automatiser la rétro-ingénierie de code Java vers diagramme de classe UML ?

Exercice 4

Question 1 • Donnez un diagramme de classes correspondant au code source suivant :

```

1  public interface Dessinable {
2  public void dessiner ( );
3  public void effacer ( );
4  }

```

```

1  abstract public class Figure implements Dessinable {
2  protected String couleur;
3  protected String getCouleur ( ) { return couleur; }
4  protected void setCouleur ( String c ) { couleur = c; }
5  }

```

```

1  public class Point {
2  private float x;
3  private float y;
4  public float getX ( ) { return x; }
5  public float getY ( ) { return y; }
6  public void Point ( float x, float y) { ... }
7  }

```

```

1  public class Cercle extends Figure {
    private float rayon;
3   private Point centre;
    public Cercle ( Point centre, float rayon) { ... }
5   public void dessiner ( ) { ... }
    public void effacer ( ) { ... }
7   }

```

```

1  public class Rectangle extends Figure {
    protected Point sommets[] = new Point[2];
3   public Rectangle ( Point p1, Point p2) { ... }
    public void dessiner ( ) { ... }
5   public void effacer ( ) { ... }
    }

```

```

    public class Losange extends Figure {
2     protected Point sommets[] = new Point[2];
    public Losange ( Point p1, Point p2) { ... }
4     public void dessiner ( ) { ... }
    public void effacer ( ) { ... }
6     }

```

Exercice 5

Question 1 • Donnez un diagramme de séquence correspondant au code source suivant :

```

1  public class Account {
3     private int _id;
    private String _name;
5     private int _age;

7     public void setId(int id){ _id = id; }
    public int getId(){ return _id; }
9     public void setName(String name){ _name = name; }
    public String getName(){ return _name; }
11    public void setAge(int age){ _age = age; }
    public int getAge(){ return _age; }
13    }

```

```
1  import java.util.*;
3
3  public class RegisterController {
5
5  private List _accounts = new ArrayList();
7
7  public void register(String name, int age) {
8      Account account = new Account();
9      account.setId(1);
10     account.setName(name);
11     account.setAge(age);
12     _accounts.add(account);
13 }
14
15 public List getAccounts(){ return _accounts;
16 }
17 }
```