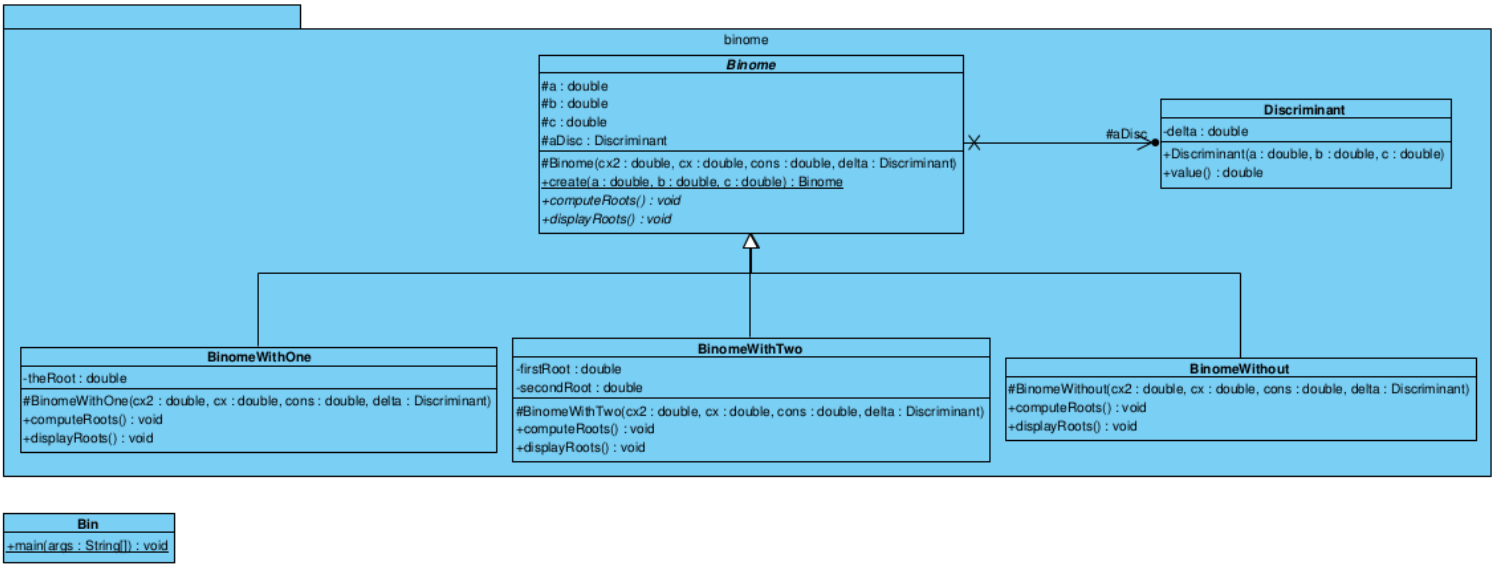
M3105 : TP4
Antoine Gicquel

Question 1 :
A l'aide du code ci-dessous :
a)Télécharger l'archive du code et construire le diagramme de classes de conception en conservant les packages;

Le diagramme :

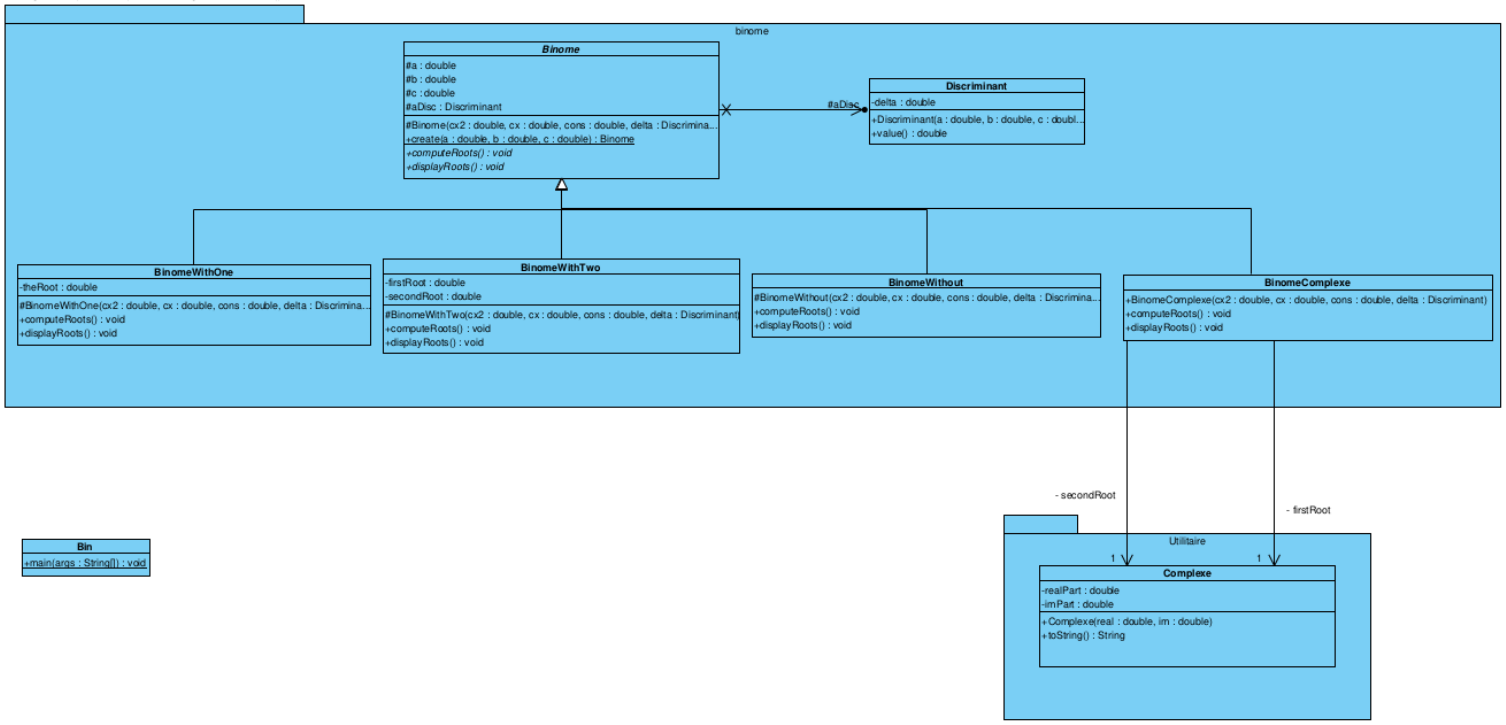


b) puis expliquer le statut de la méthode create;
La méthode create de l'objet Binome est statique et renvoie un objet hérité de Binome. Binome suit le patron de conception « abstract factory » et va créer l'objet adapté en fonction des paramètres qui lui sont fournis.

c) Enfin pourquoi le constructeur Binome est déclaré protected ?
Il est déclaré protected pour qu'uniquement les classes de son package aient accès à son constructeur et ainsi forcer la main au utilisateur de la classe Binome à utiliser la méthode create.

Question 3 :
Modifier le diagramme de classes de conception pour qu'il tienne compte de binômes à racines complexes en faisant en sorte que l'on puisse réutiliser dans une autre application le concept de Complexe. Cela signifie qu'il faut définir une classe Complexe qui représente un nombre complexe et sa manipulation.
Modifier le code en conséquence et faire tourner votre programme.

Le diagramme :



Le code :

```
package binome;

import Utilitaires.*;

/**
 *  Binome object whithout roots
 */

class BinomeComplexe extends Binome {
  private Complexe firstRoot;
  private Complexe secondRoot;

    protected BinomeComplexe ( double cx2,double cx, double cons, Discriminant
delta) {
        super (cx2, cx, cons, delta);
    }

    public void computeRoots() {
        this.firstRoot = new Complexe((-b / 2.0 * a ),
            (- Math.sqrt(-aDisc.value())) / (2.0 * a));
        this.secondRoot = new Complexe((-b / 2.0 * a ),
            ( Math.sqrt(-aDisc.value())) / (2.0 * a));
    }

    public void displayRoots() {
        System.out.println ("Deux racines distinctes : \n\tx1 = "
            + firstRoot.getReal() + " + "
            + firstRoot.getIm()  + "i");
        System.out.println ("\tx2 = " + secondRoot.getReal()
            + " + " + secondRoot.getIm()  + "i");
    }
}

package Utilitaires;

public class Complexe {
  private double realPart;
  private double imPart;

  public Complexe(double real, double im) {
    this.realPart = real;
    this.imPart = im;
  }

  public double getReal() {
    return this.realPart;
  }

  public double getIm() {
    return this.imPart;
  }

  public String toString() {
    String info = "";
    info += "\nThe real part : " + this.realPart;
    info += "\nThe imaginary part : " + this.imPart;
    return info;
  }
}
```