

- (10) 1. Écrire la méthode Java

```
private void remplirRectangle(Graphics g, int x,int y,int l,int h)
```

qui remplit un rectangle de largeur l et hauteur h dont le coin gauche supérieur se trouve à x,y comme le ferait la méthode `fillRect` de la classe `Graphics`, mais en n'utilisant que ses méthodes `drawRect(int x,int y,int l,int h)` et/ou `drawLine(int x1,int y1, int x2, int y2)` qui dessinent respectivement un rectangle et une ligne.

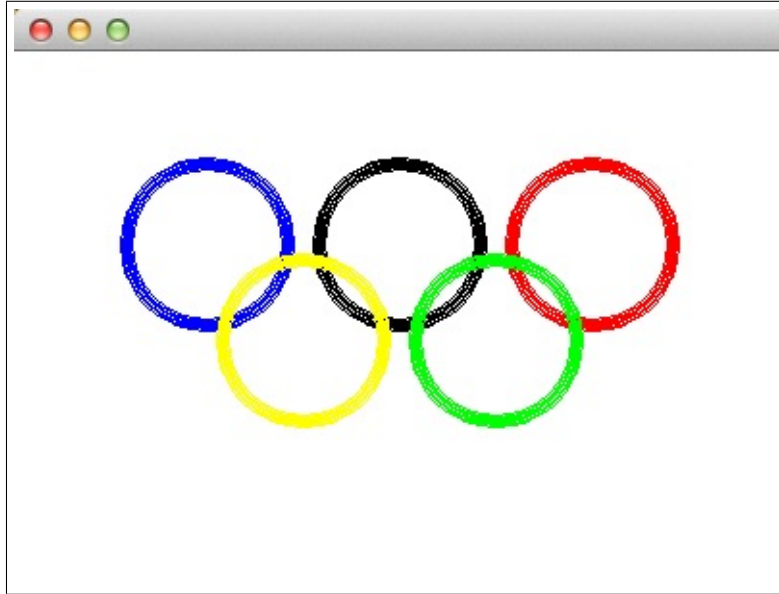


Figure 1: Approximation du sigle des jeux olympiques

- (40) 2. Définissez une classe `Anneau` qui représente à la position x et y un anneau de rayon r et de largeur l . **Réutilisez autant que possible** les classes `Baloune` et `Coloree` dont des extraits sont donnés en appendice (vous n'avez pas à tenir compte des méthodes présentées en cours, mais qui ne sont pas décrites en appendice).
- (5) (a) Écrivez une classe qui définit une exception `AnneauIllegalException`.
- (5) (b) Définissez le constructeur de la classe `Anneau`. Ce constructeur devrait lever l'exception `AnneauIllegalException` si le rayon ou la largeur de l'anneau sont négatifs ou nuls ou si la largeur est plus grande que le rayon.
- (10) (c) Définissez la méthode

```
public void afficher(Graphics g)
```

qui dessine l'anneau dans un contexte graphique g . Notez qu'il ne faut pas que l'intérieur de l'anneau soit rempli.

- (20) (d) Utilisez la classe Anneau pour dessiner une approximation du sigle des jeux olympiques (voir Figure 1) avec cinq anneaux de couleurs différentes (ils n'ont pas à être entrelacés comme le sigle officiel) en complétant le gabarit de code suivant où vous aurez complété les parties indiquées par `/** X */`. La partie A vaut 10 points et les parties B et C en valent 5 chacune.

```
class OlympiquePanel extends JPanel {
    final int rayonAnneau    = 45;
    final int largeurAnneau = 5;
    // couleurs des 5 anneaux
    Color cs[]=new Color[]{Color.blue,Color.black,Color.red,
                          Color.yellow,Color.green};
    int xs[] = new int[]{100,200,300,150,250}; // coordonnées X des anneaux
    int ys[] = new int[]{100,100,100,150,150}; // coordonnées Y des anneaux

    Anneau[] as;

    OlympiquePanel(){
        /** A (10 points) */
    }

    @Override
    public void paintComponent(Graphics g){
        super.paintComponent(g);
        /** B (5 points) */
    }
}

public class Olympique {
    public static void main(String[] args) {
        JFrame f = new JFrame();
        /** C (5 points) */
        f.setSize(400,300);
        f.setLocation(100,100);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



Figure 2: Application de gonflage de **Face**

- (50) 3. Complétez la classe `FaceEnfilee` esquissée ci-bas pour implanter un jeu simple (voir Figure 2) qui affiche une instance de la classe `Face` dans une fenêtre avec deux boutons qui la gonflent et la dégonflent par coup de 10 pixels à chaque appui sur le bouton. Lorsqu'on clique sur la `Face`, il *change d'air*, mais ne fait rien si on clique sur l'écran mais en dehors de la face.

Pour l'implantation, utiliser la structure de programme suivante, où vous aurez complété les parties indiquées par `/** X */`. La partie A vaut 20 points et les parties B,C et D en valent 10 chacune.

Vous pouvez aussi écrire des méthodes privées auxiliaires pour simplifier votre code.

```
public class FaceEnflee {
    // déclarer les objets

    /** A (20 points) */

    private void creerObjets(){
        /** B (10 points) */
    }

    private void placerObjets(Container c){
        /** C (10 points) */
    }

    private void ecouterObjets(){
        /** D (10 points) */
    }

    FaceEnflee(Container c){
        creerObjets();
        placerObjets(c);
        ecouterObjets();
    }

    public static void main(String[] args) {
        JFrame f = new JFrame();
        new SoufflageBaloune(f.getContentPane());
        f.setSize(300,300);
        f.setLocation(100,100);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```

- (2) 4. **Question bonus.** Qu'imprime ce programme Java ? expliquez votre réponse.

```
public class Egalite {
    public static void main(String args[]){
        String s1="IFT1025";
        String s2="IFT"+1025;
        System.out.println("s1 == s2:" + s1==s2);
    }
}
```

Appendice: Rappels

```
class Baloune{
    protected int x,y,rayon; // coordonnées du centre
    private int r2,diametre;

    Baloune(int x,int y, int rayon){
        if (rayon>0) {
            this.x=x; this.y=y; this.rayon=rayon;
            diametre = 2*rayon;
            r2=rayon*rayon;
        }
    }

    public boolean gonfler(int r){
        // augmente le rayon si la nouvelle valeur reste positive
        // important car r peut être négatif
        int r1 = rayon+r;
        if (r1>0){ // on garantit que le nouveau rayon reste positif
            rayon =r1;
            diametre = 2*r1;
            r2 = r1*r1;
            return true;
        } else return false; // indique l'échec du gonflage
    }

    public boolean degonfler(int r){
        return gonfler(-r);
    }

    public void afficher(Graphics g,String mess){
        // dessine la baloune avec un message au centre
        g.drawOval(x-rayon,y-rayon,diametre,diametre);
        g.drawString(mess,x,y);
    }

    private int sqr(int x){ return x*x;}

    public boolean contient(int px, int py){
        // vérifie si le point px@py est dans la baloune
        return sqr(px-x)+sqr(py-y) <= r2;
    }
}
```

```
class Coloree extends Baloune{
    protected Color couleur;

    Coloree(int x, int y, int rayon, Color couleur){
        super(x,y,rayon);
        this.couleur=couleur;
    }

    public void afficher(Graphics g, String mess){
        // remplit la couleur avant de faire dessiner le tour et le message en noir
        g.setColor(couleur);
        g.fillOval(x-rayon,y-rayon,2*rayon,2*rayon);
        g.setColor(Color.black);
        super.afficher(g,mess);
    }
}

class Face extends Coloree{
    // Baloune colorée mais qui s'affiche comme triste ou gaie
    private boolean triste;

    Face(int x, int y, int rayon, Color couleur){
        super(x,y,rayon,couleur);
    }

    public void afficher(Graphics g, String mess){
        super.afficher(g,"");
        // affiche les yeux et le nez
        int r2=rayon/2, r3=rayon/3, r6=rayon/6;
        g.fillOval(x-r3-r6/2,y-r3,r6,r6);
        g.fillOval(x+r3,y-r3,r6,r6);
        g.fillOval(x-r6/2,y-r6/2,r6,r6);
        if (triste) g.drawArc(x-r2,y+r2,rayon,rayon,50,80); // la moue
        else g.drawArc(x-r2,y-r2,rayon,rayon,230,80); // le sourire
        // affiche le nom en bas
        g.drawString(mess,x,y+rayon+12);
    }

    public void changeDAir(){
        triste = !triste;
    }
}
```