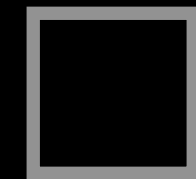


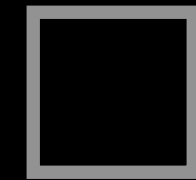
Graphiques



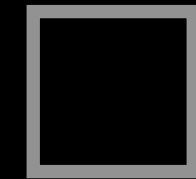
Objectif de ce cours



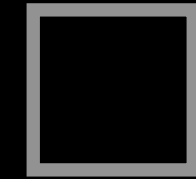
Variables, révision



GraphicsProgram: Introduction et exemple



Quelques objets graphiques communs



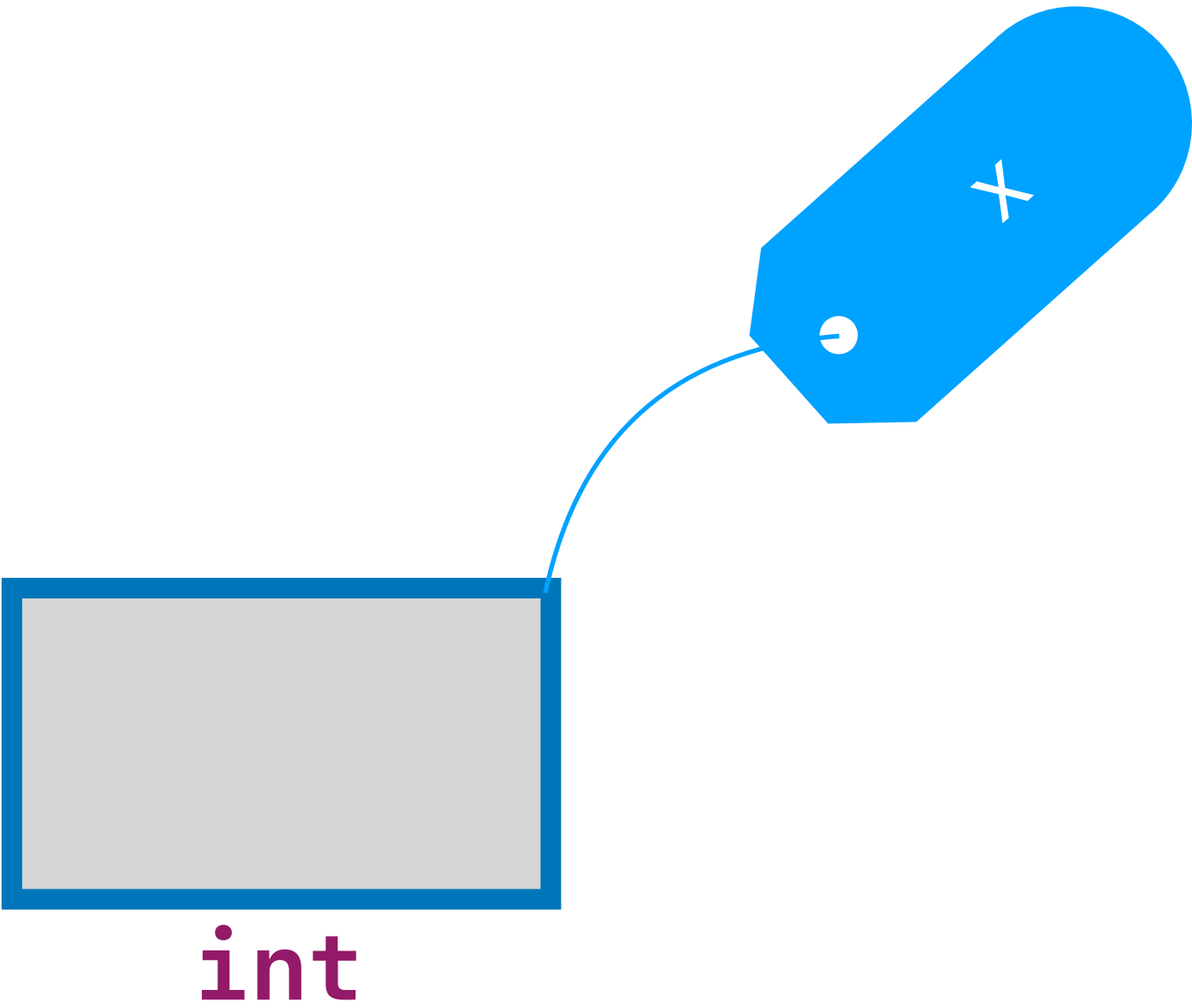
Grand-père de Karel

Variables, révision: Déclaration et affectation

```
int x;
```

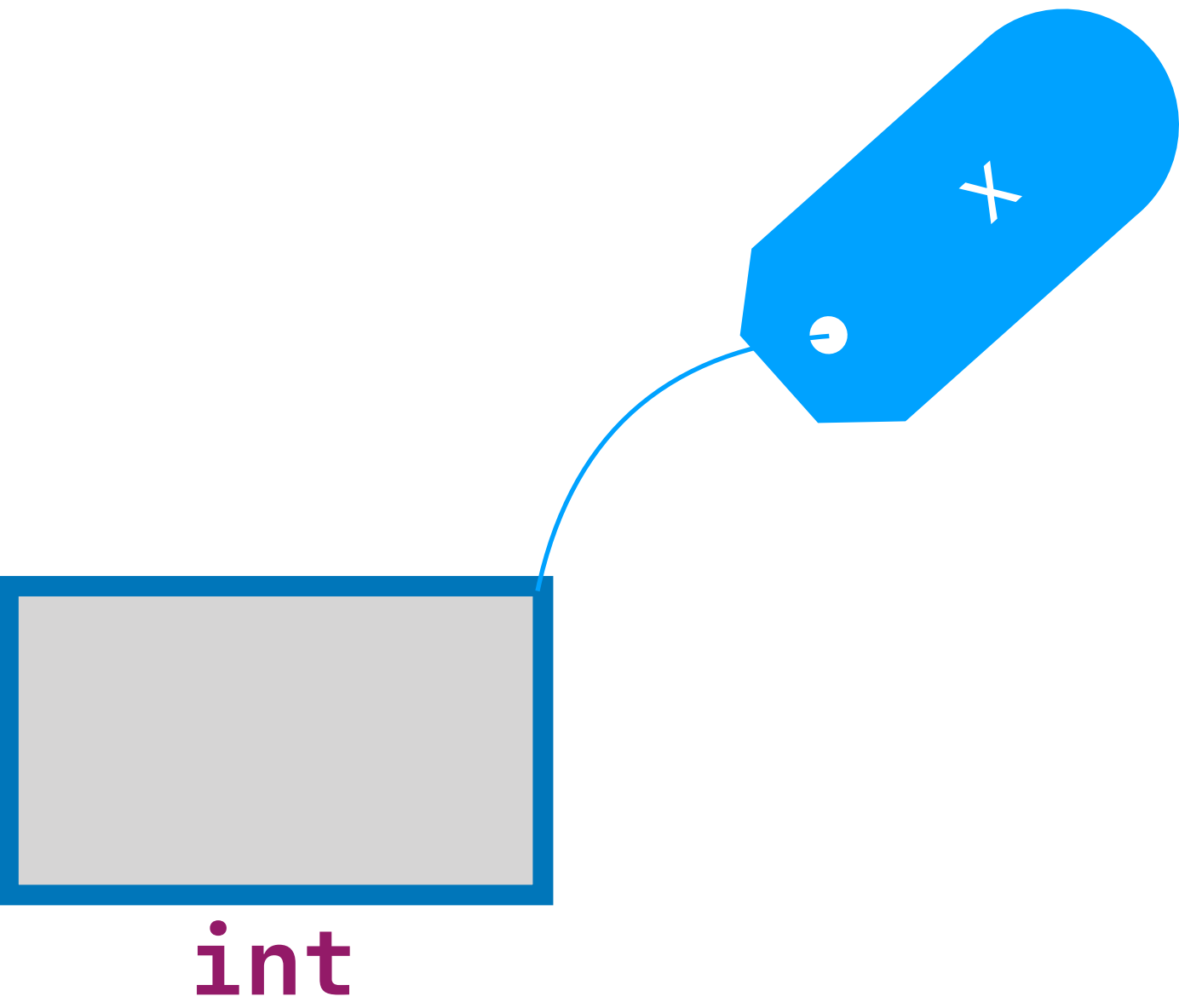
Variables, révision: Declaration et affectation

```
int x;
```



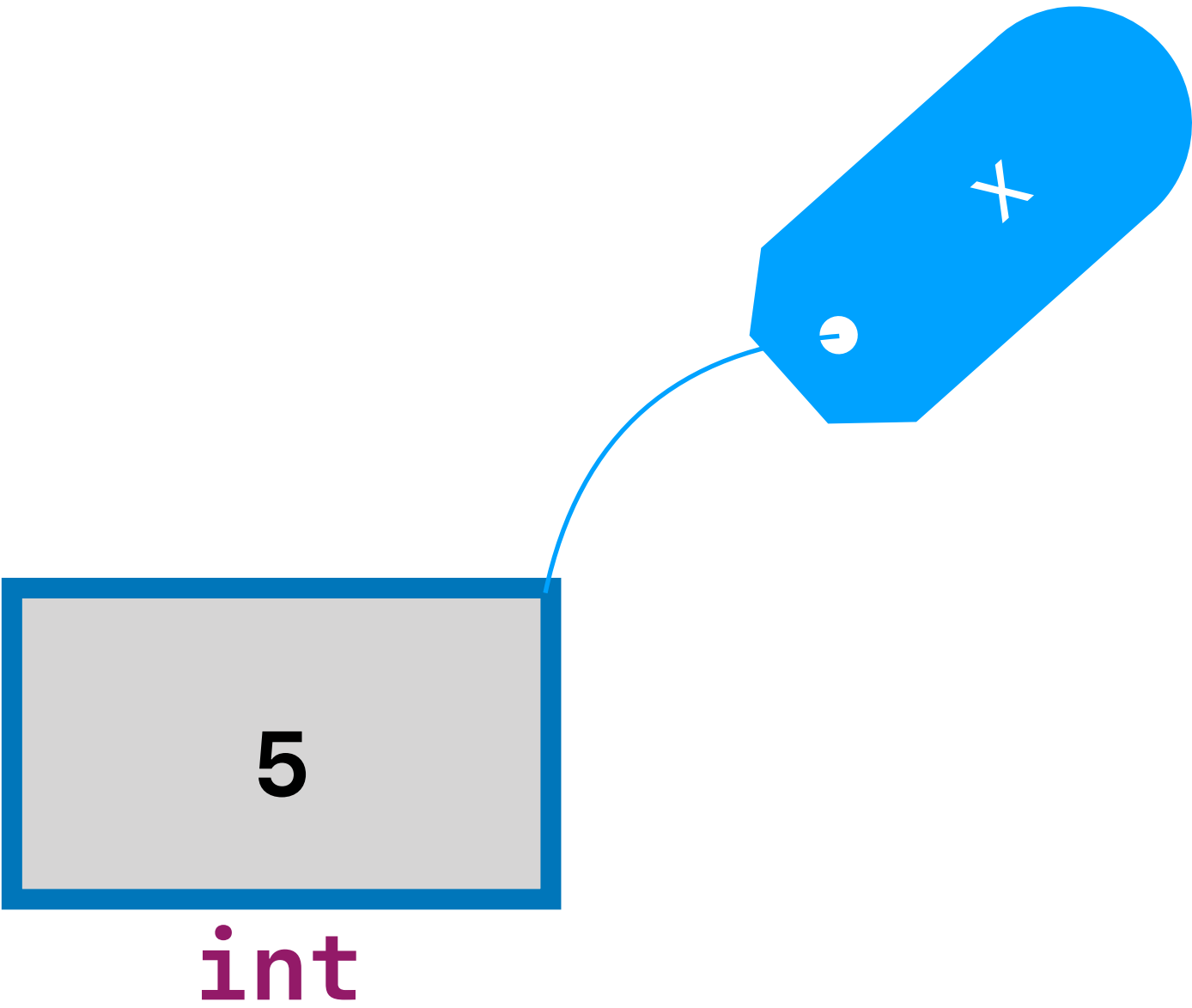
Variables, révision: Declaration et affectation

```
int x;  
x = 5;
```



Variables, révision: Declaration et affectation

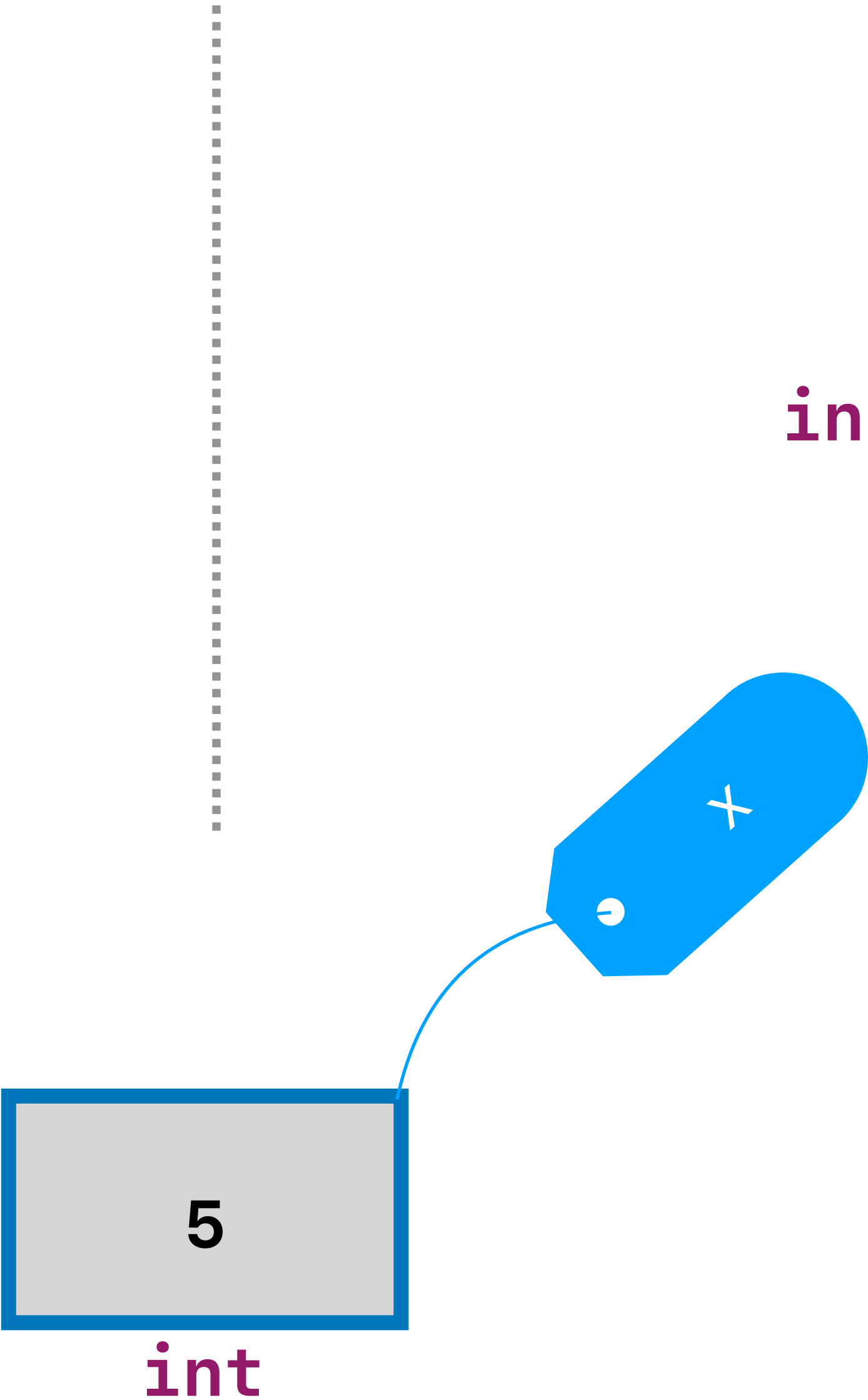
```
int x;  
x = 5;
```



Variables, révision: Declaration et affectation

```
int x;  
x = 5;
```

```
int x = 5;
```



Variables, révision: Types

Variables, révision: Types

La premiere lettre d'un prénom

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

```
int nbEleves = 10;
```


Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

```
int nbEleves = 10;
```

L'eleve est-il-admis?

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

```
int nbEleves = 10;
```

L'eleve est-il-admis?

```
boolean estAdmis = true;
```

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

```
int nbEleves = 10;
```

L'eleve est-il-admis?

```
boolean estAdmis = true;
```

Le nom d'un élève

Variables, révision: Types

La premiere lettre d'un prénom

```
char premiereLettre = 'M';
```

La taille d'un élève

```
double taille = 1.5;
```

Nombre d'élèves dans une classe

```
int nbEleves = 10;
```

L'eleve est-il-admis?

```
boolean estAdmis = true;
```

Le nom d'un élève

```
String nom = "Lucien";
```

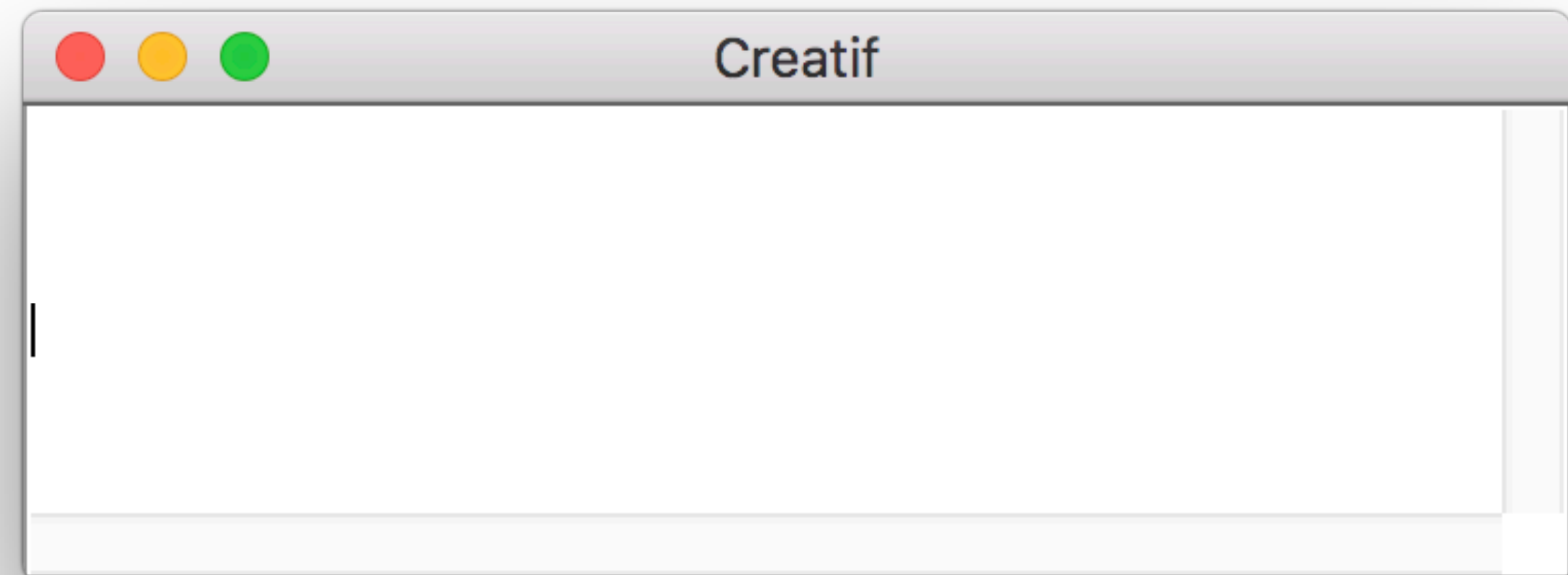
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



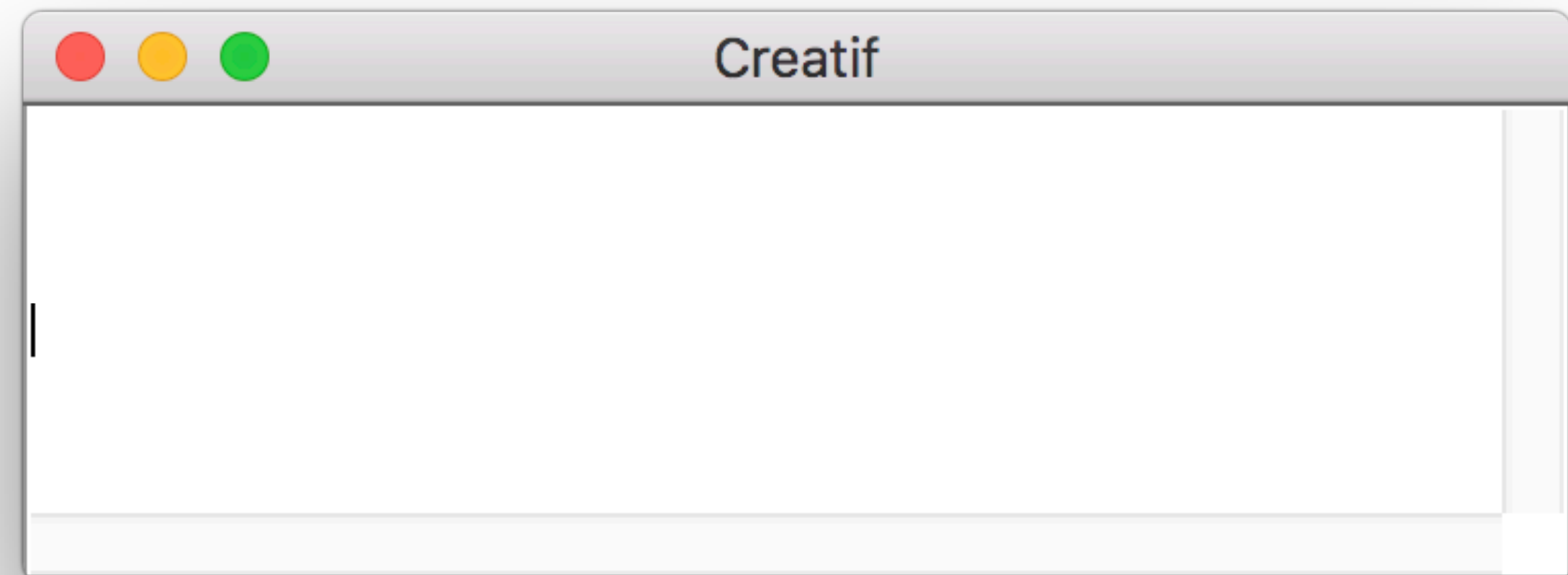
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



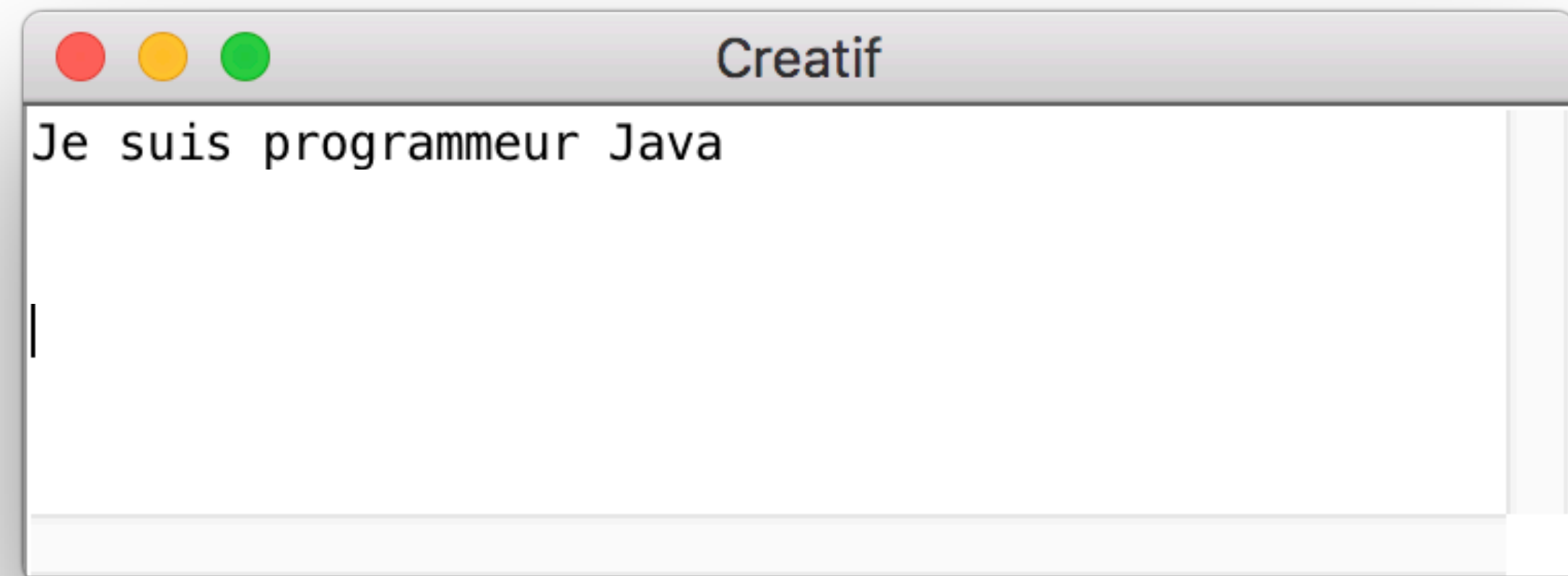
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



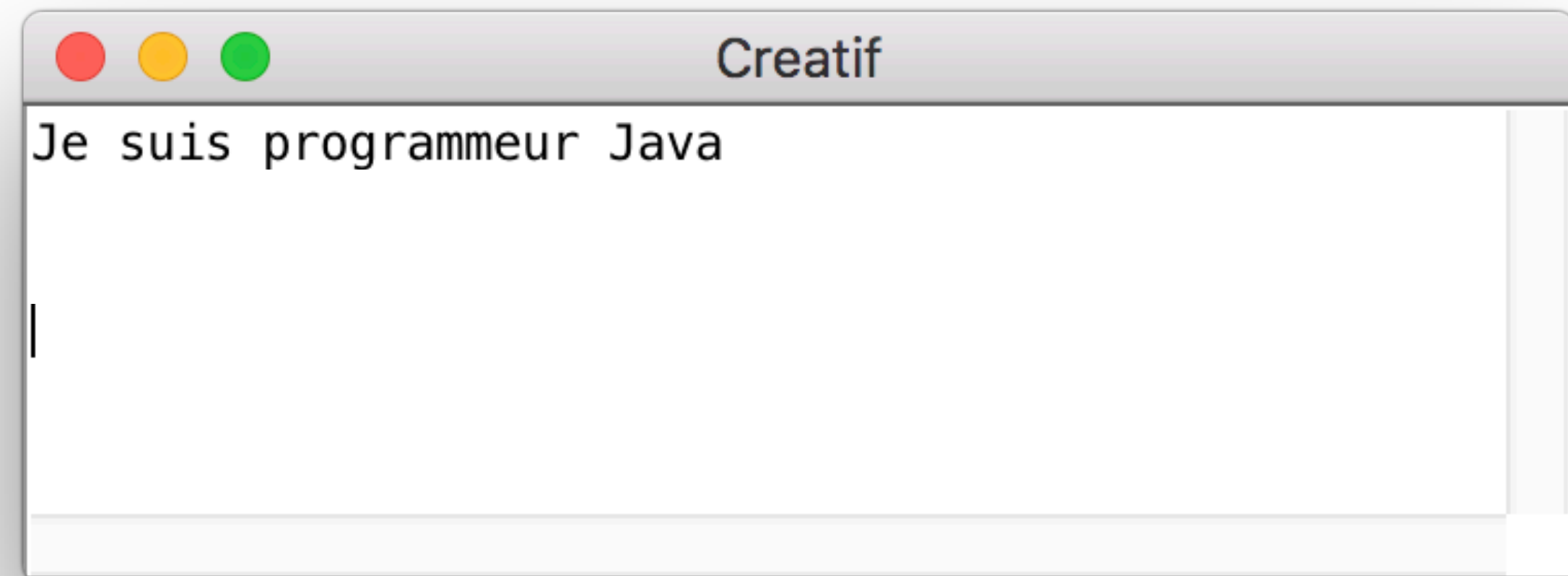
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



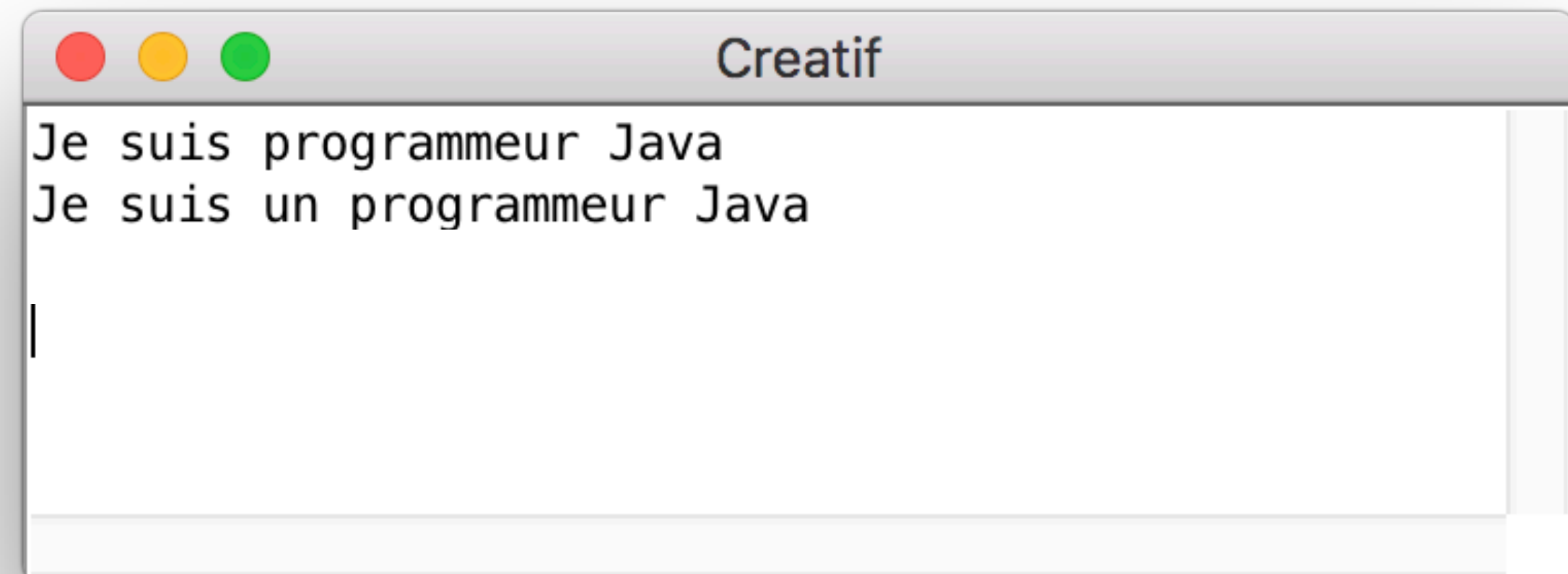
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



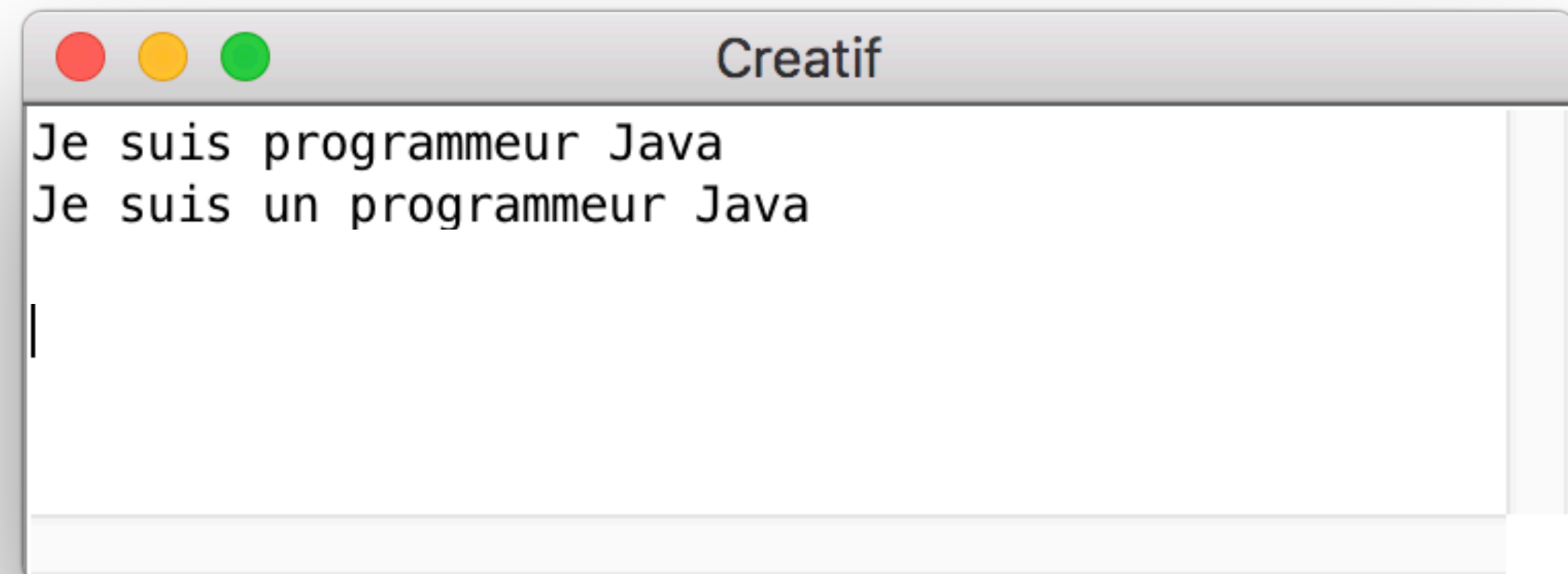
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```



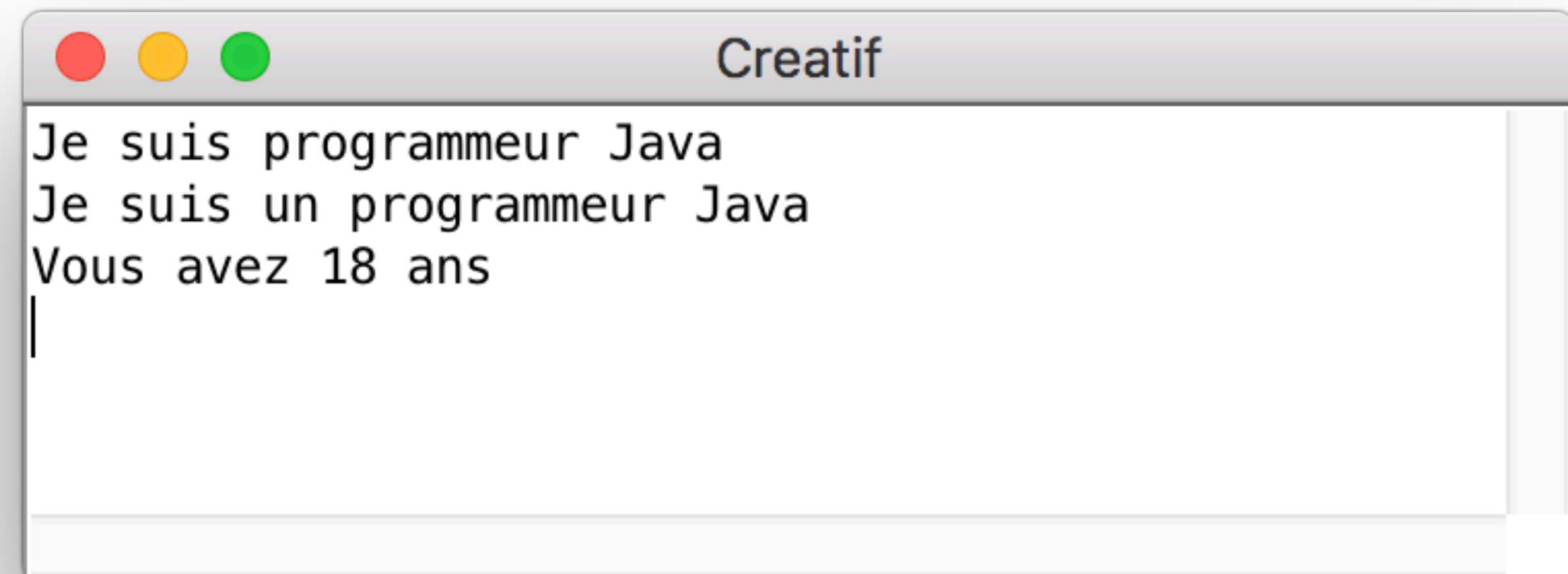
Variables, révision: Chaines de caractères

```
String s1 = "Je suis";  
String s2 = " un";  
String s3 = " programmeur Java";
```

```
String s4 = s1 + s3;  
println(s4);
```

```
String s5 = s1 + s2 + s3;  
println(s5);
```

```
int age = 18;  
String s6 = "Vous avez " + age + " ans";  
println(s6);
```

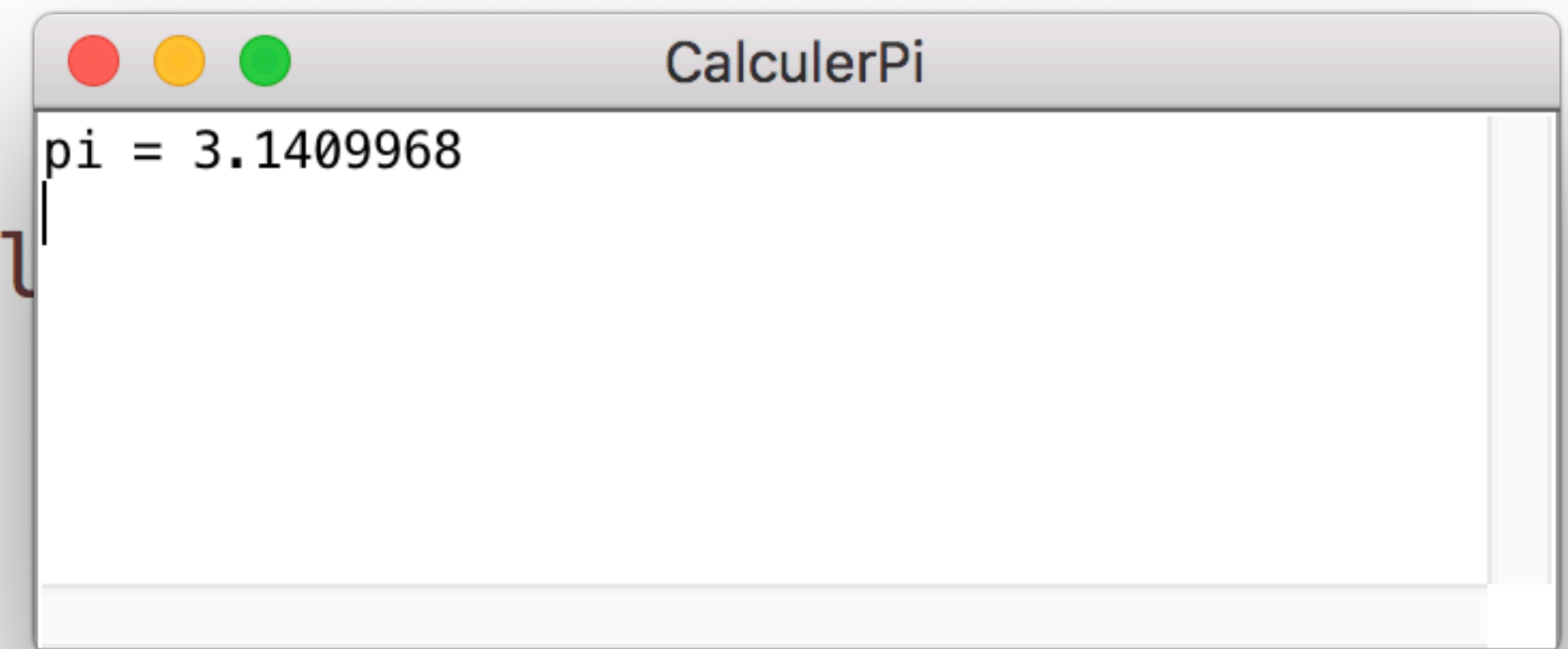


Variables, révision: Calculer Pi

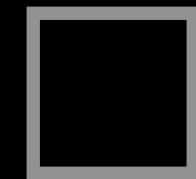
```
4 public class CalculerPi extends ConsoleProgram {
5
6     public void run() {
7         int nbFleches = 5000000;
8         RandomGenerator rg = new RandomGenerator();
9         int nbFlechesDansCercle = 0;
10        for (int i=0; i<nbFleches; i++) {
11            double x = rg.nextDouble(-1, 1);
12            double y = rg.nextDouble(-1, 1);
13            double distance = Math.sqrt(x*x + y*y);
14            if(distance < 1) {
15                nbFlechesDansCercle = nbFlechesDansCercle + 1;
16            }
17        }
18
19        double pi = 4.0*nbFlechesDansCercle / nbFleches;
20        println("pi = " + pi);
21    }
22 }
```

Variables, révision: Calculer Pi

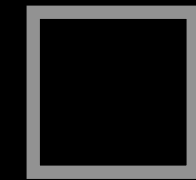
```
4 public class CalculerPi extends ConsoleProgram {
5
6     public void run() {
7         int nbFleches = 5000000;
8         RandomGenerator rg = new RandomGenerator();
9         int nbFlechesDansCercle = 0;
10        for (int i=0; i<nbFleches; i++) {
11            double x = rg.nextDouble(-1, 1);
12            double y = rg.nextDouble(-1, 1);
13            double distance = Math.sqrt(x*x + y*y);
14            if(distance < 1) {
15                nbFlechesDansCercle = nbFlechesDansCercle + 1;
16            }
17        }
18
19        double pi = 4.0*nbFlechesDansCercle/nbFleches;
20        println("pi = " + pi);
21    }
22 }
```



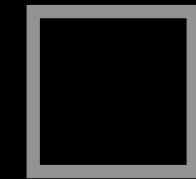
Objectif de ce cours



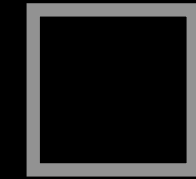
Variables, révision



GraphicsProgram: Introduction et exemple



Quelques objets graphiques communs

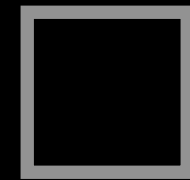


Grand-père de Karel

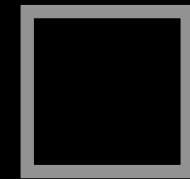
Objectif de ce cours



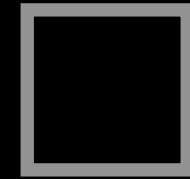
Variables, révision



GraphicsProgram: Introduction et exemple



Quelques objets graphiques communs

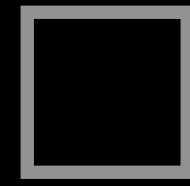


Grand-père de Karel

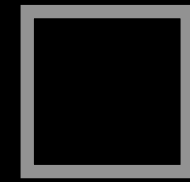
Objectif de ce cours



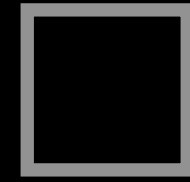
Variables, révision



GraphicsProgram: Introduction et exemple

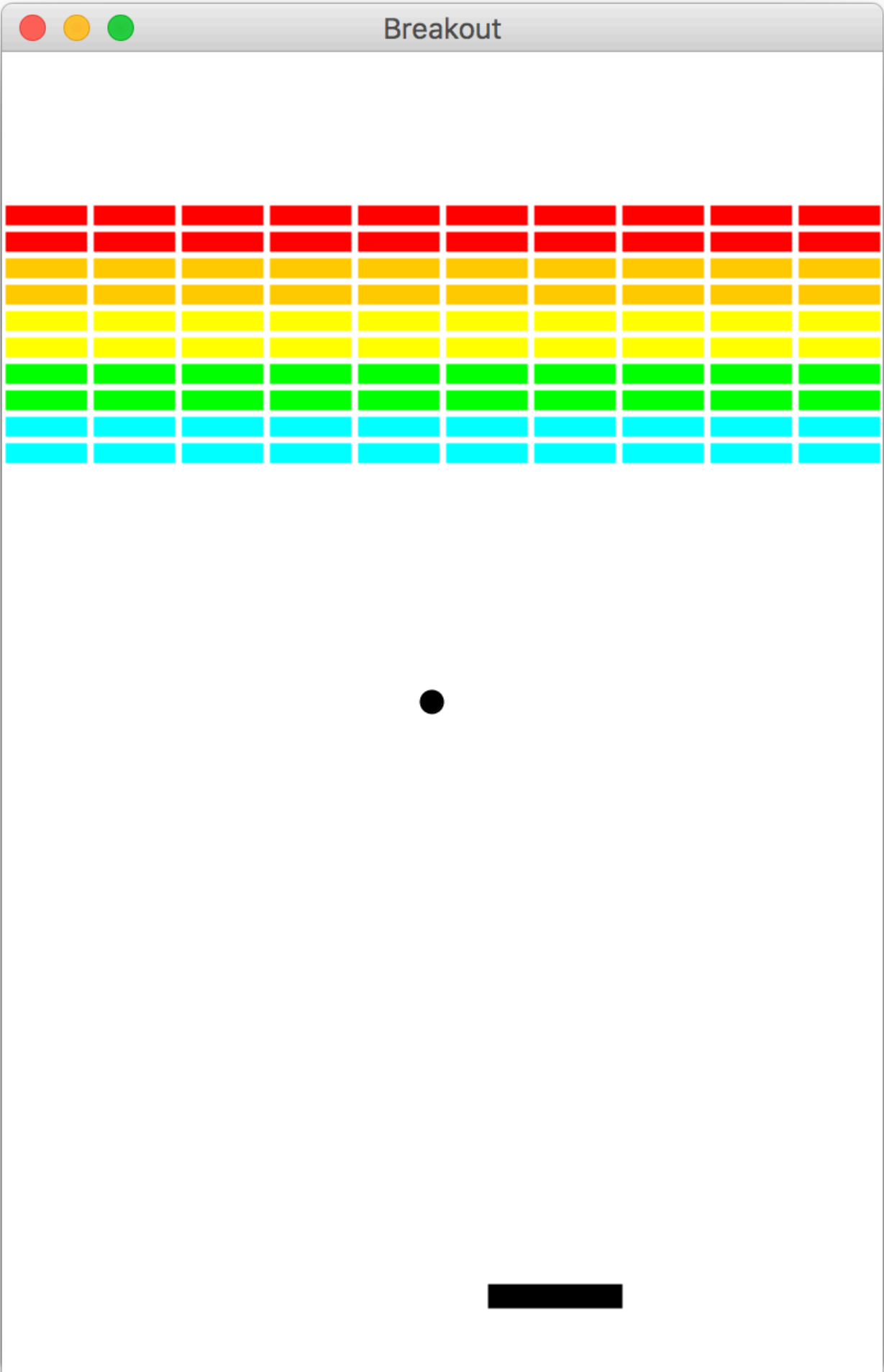
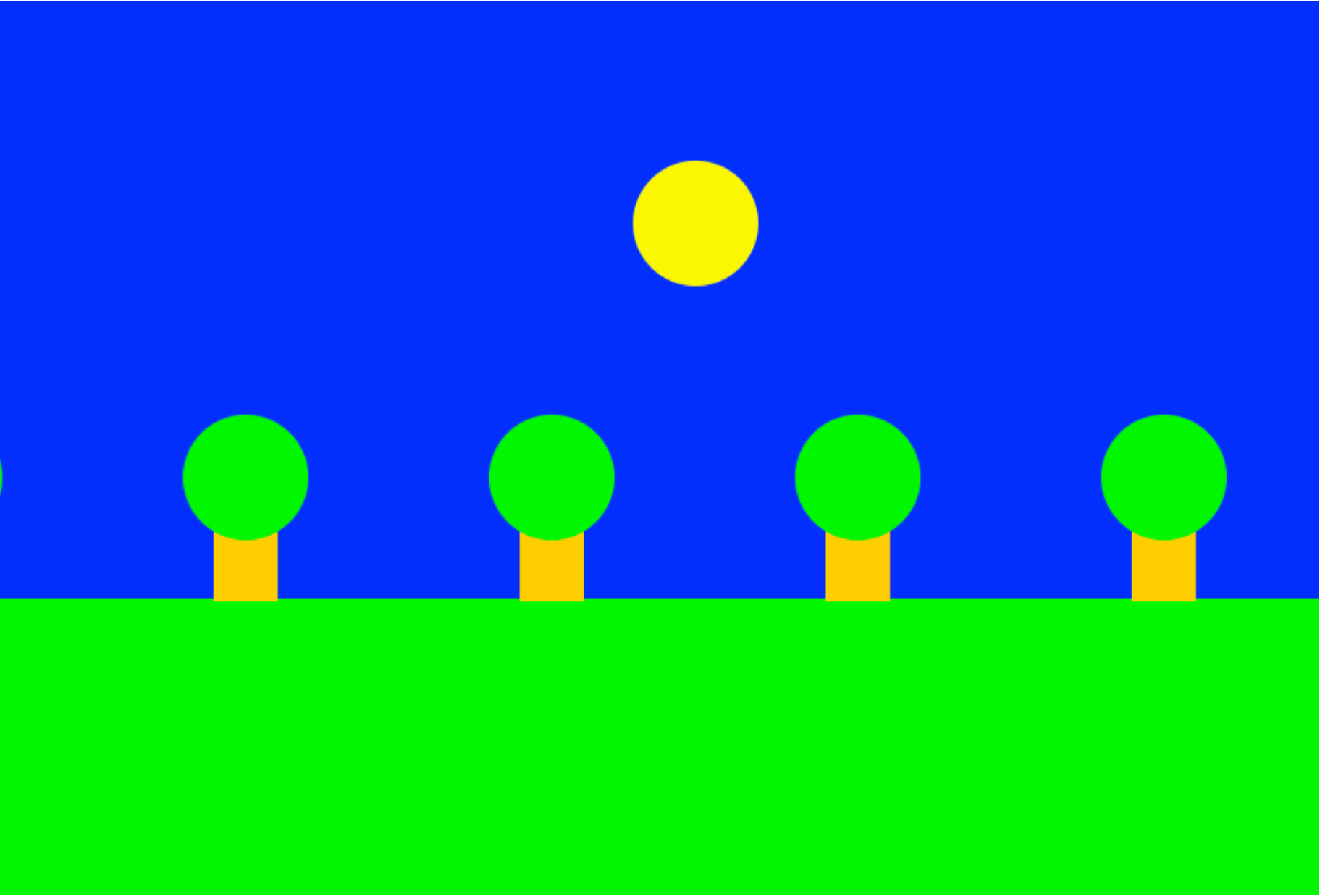
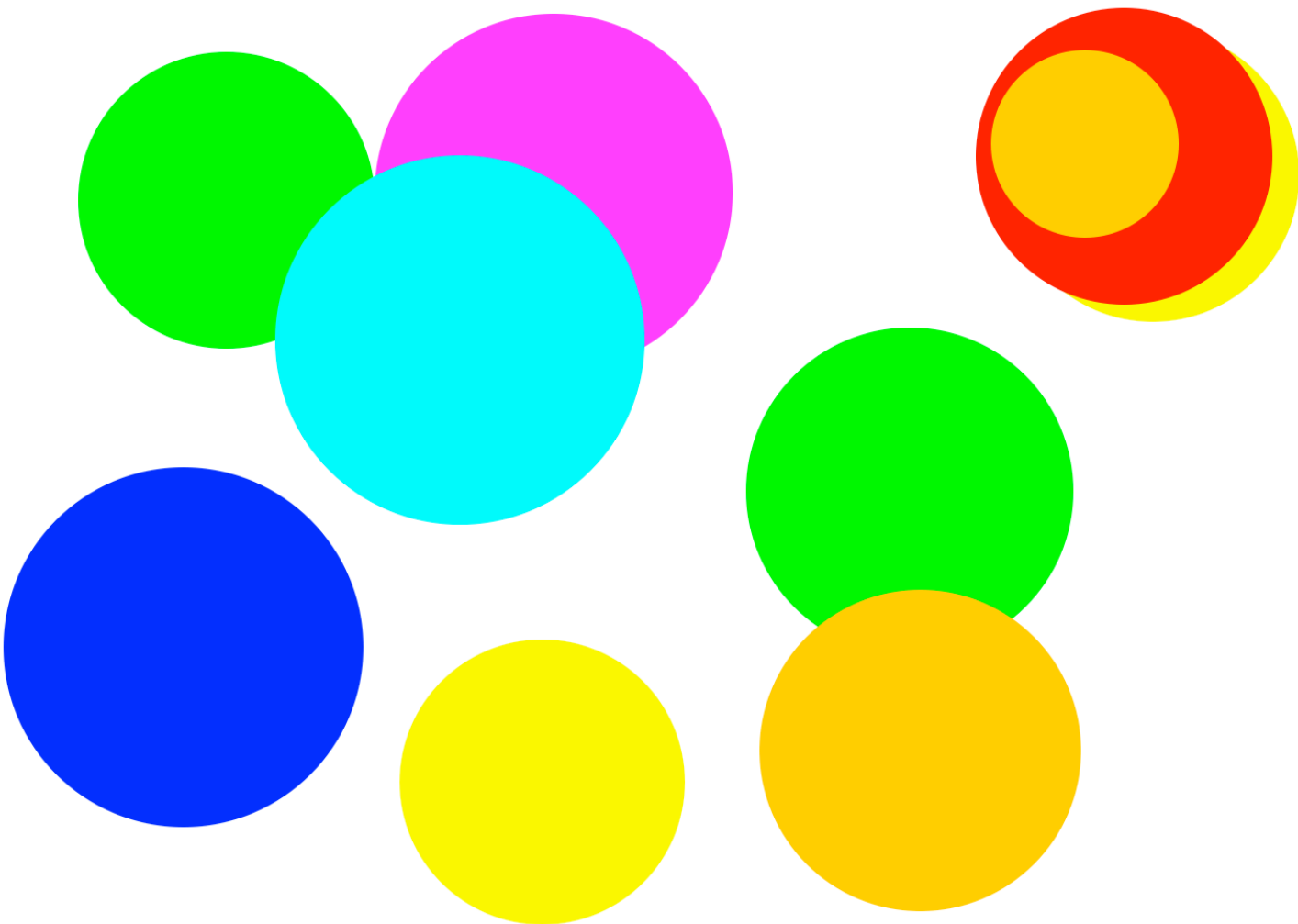


Quelques objets graphiques communs



Grand-père de Karel

GraphicsProgram



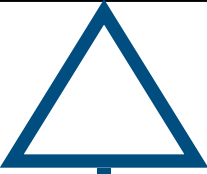
Programmes

Program

Programmes



Programmes



Programmes



Program



Karel

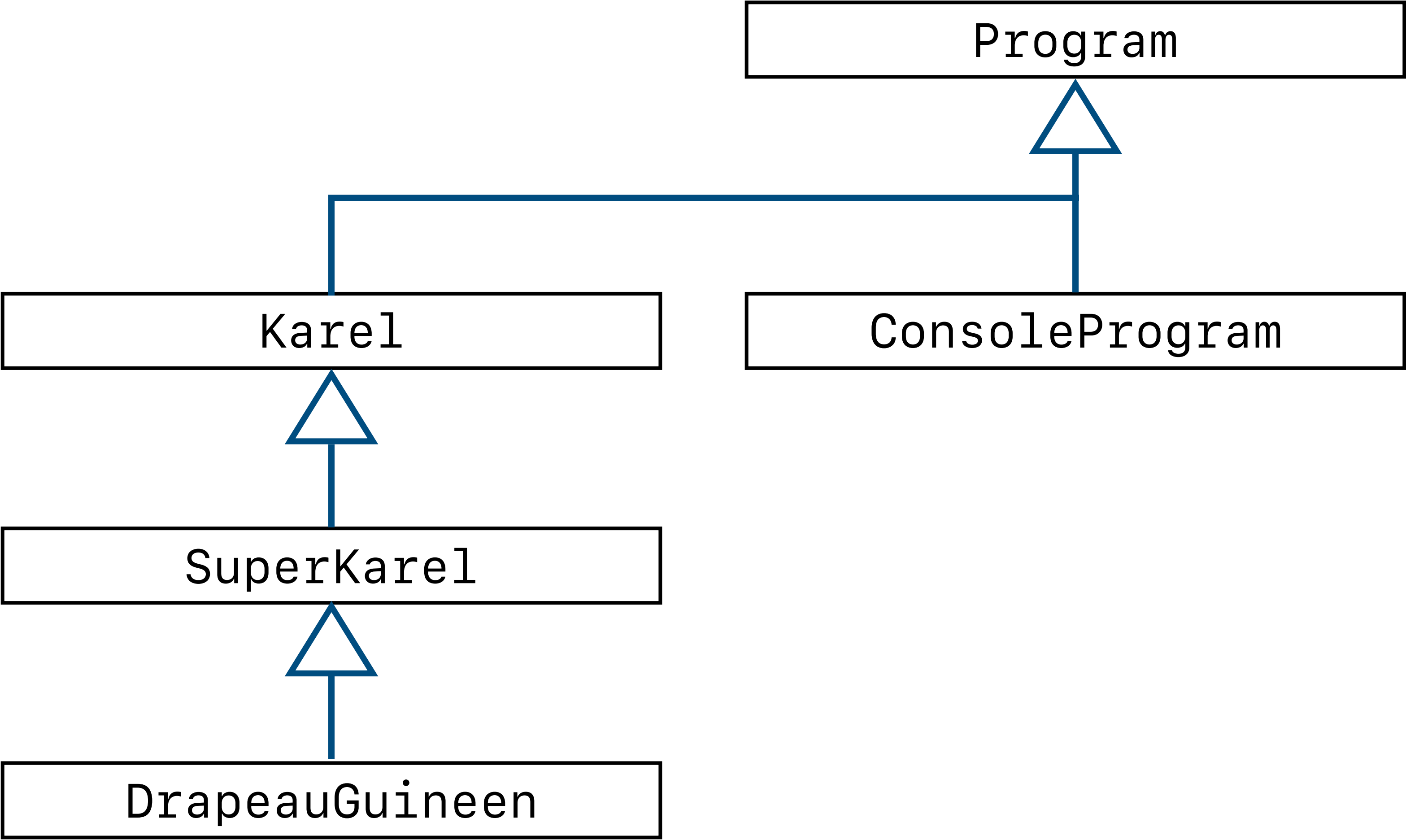


SuperKarel

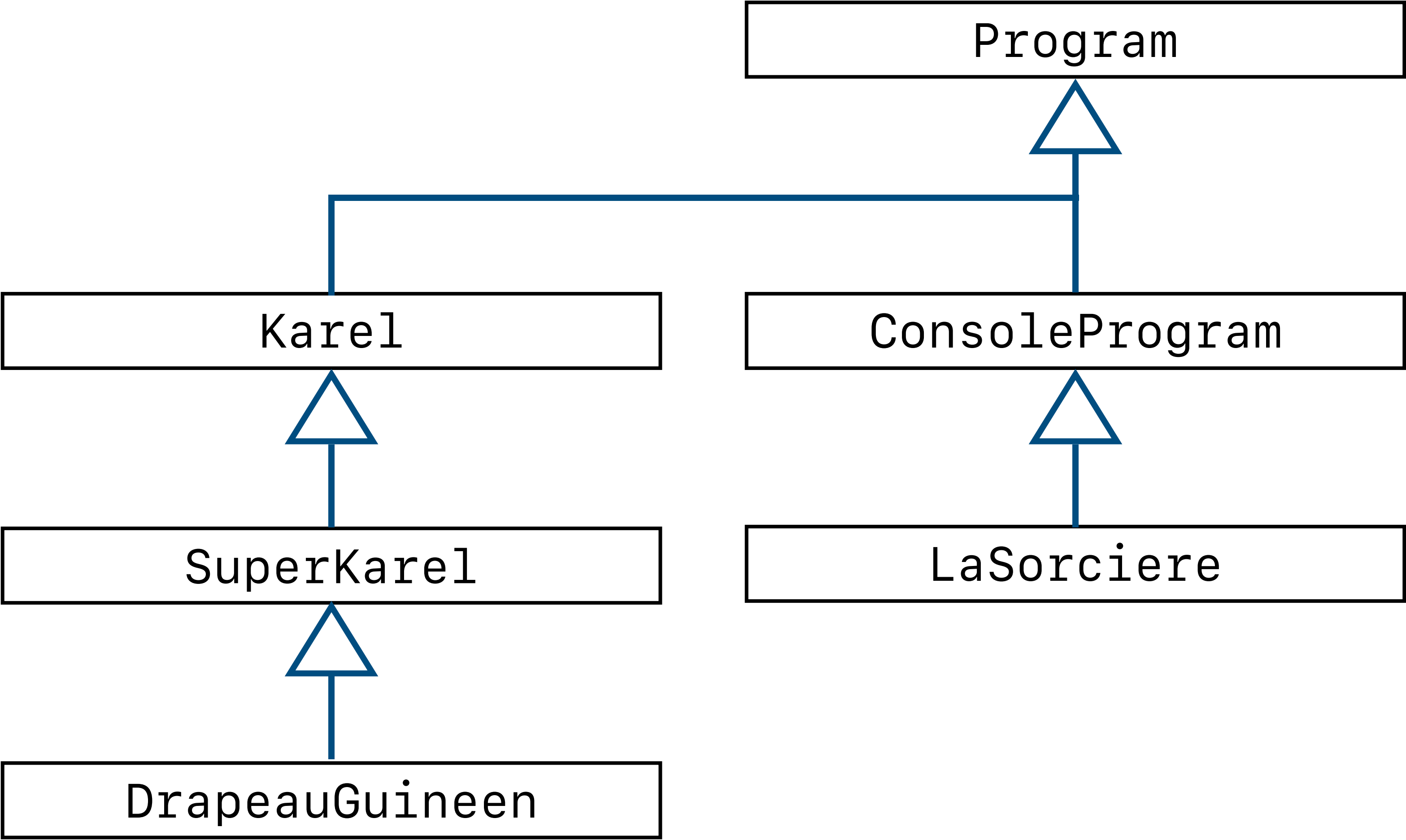


DrapeauGuineen

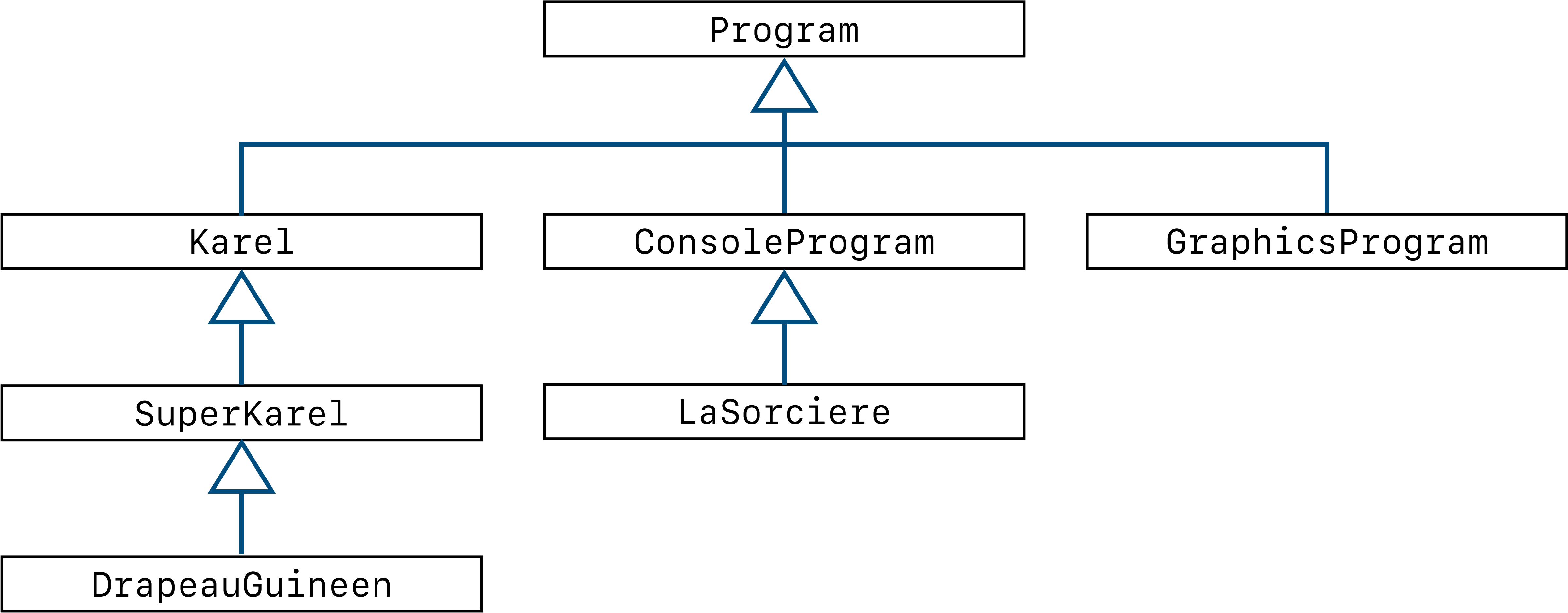
Programmes



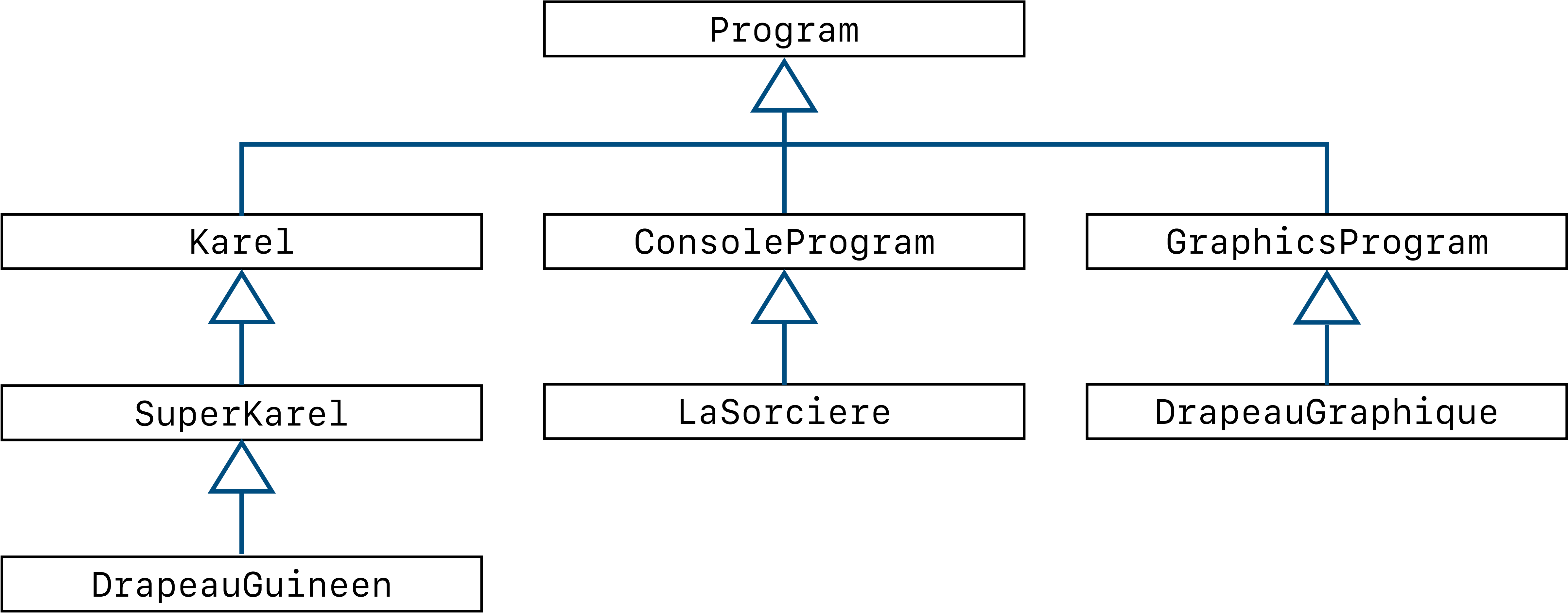
Programmes



Programmes

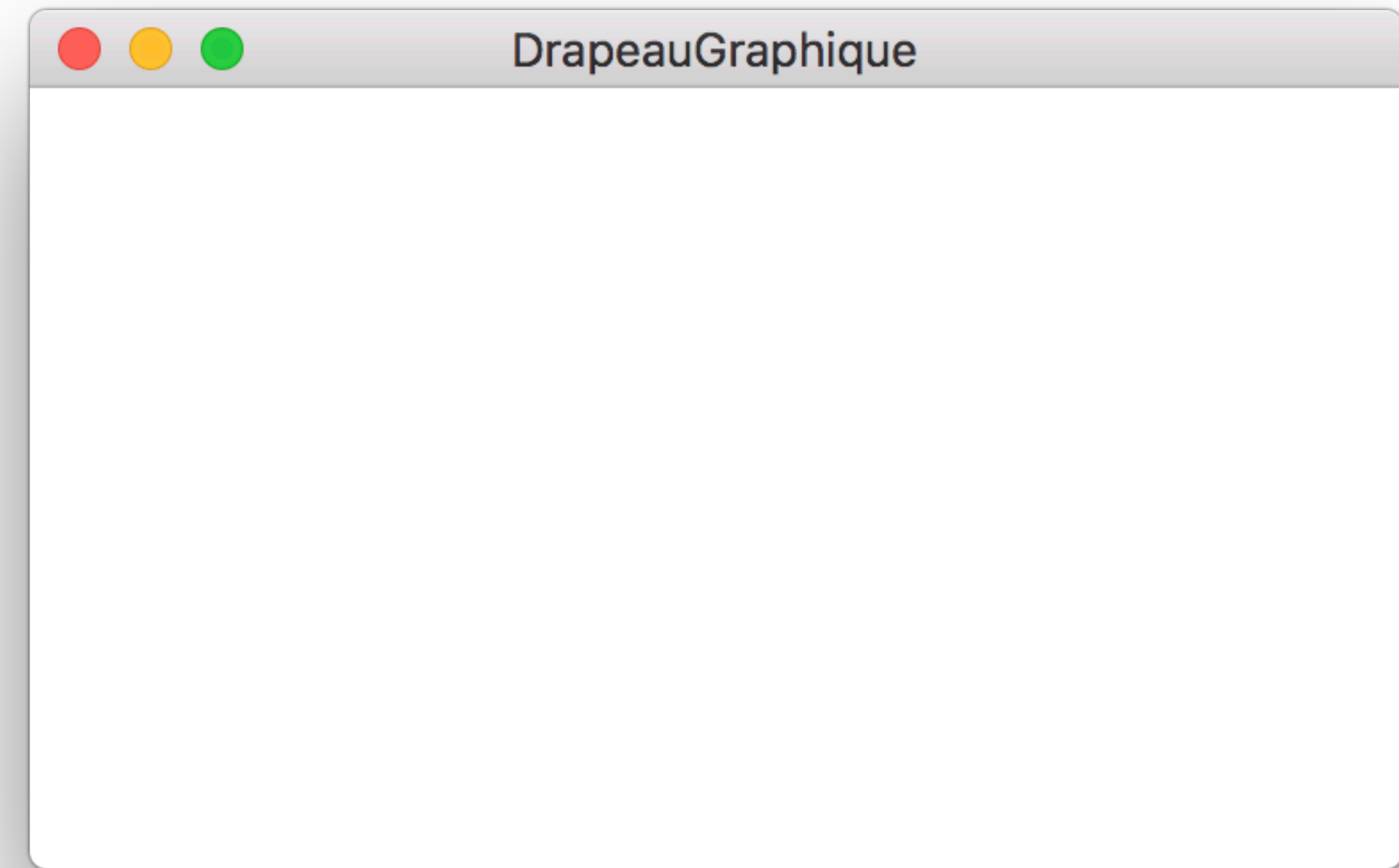


Programmes



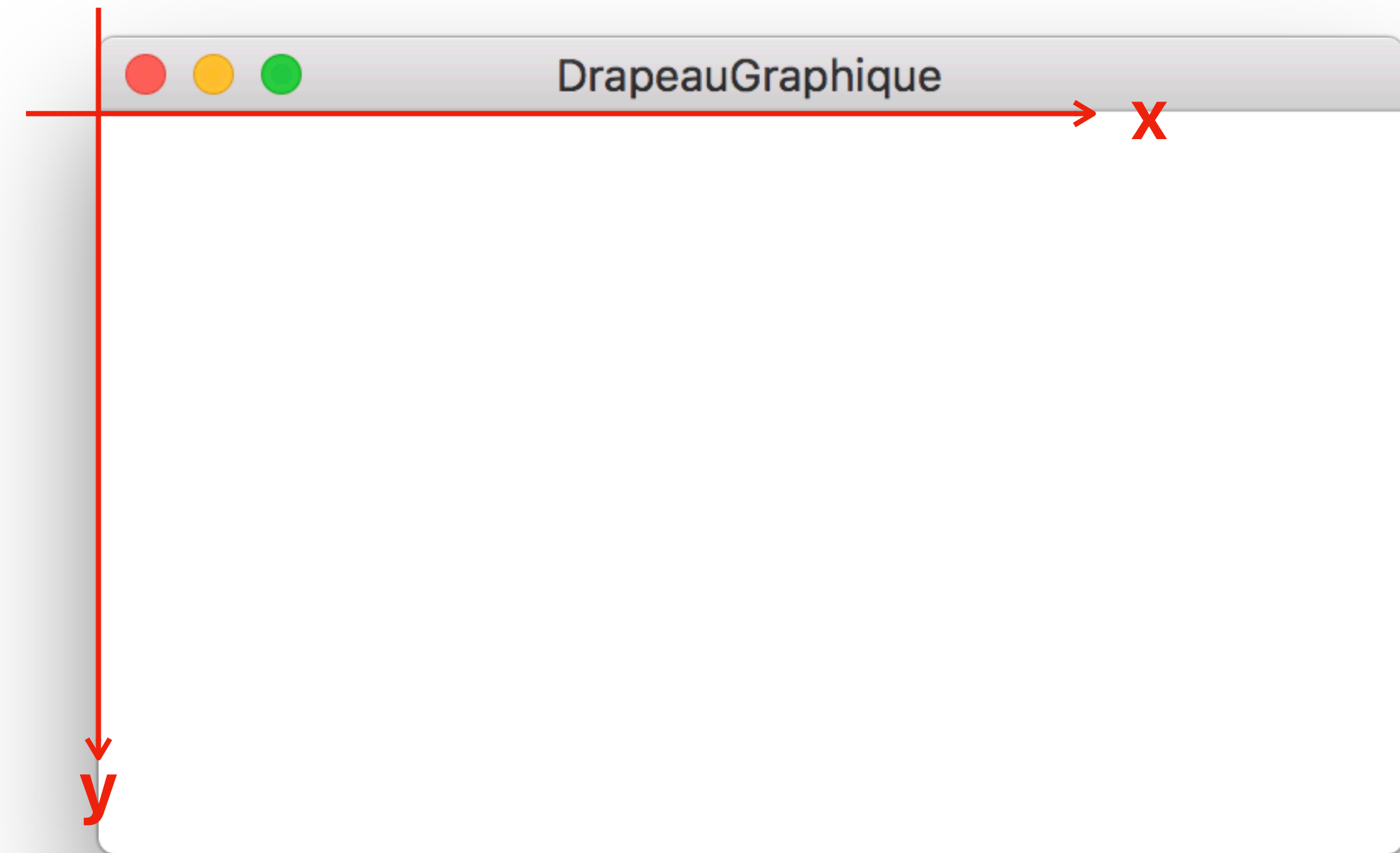
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {
    public void run() {
```



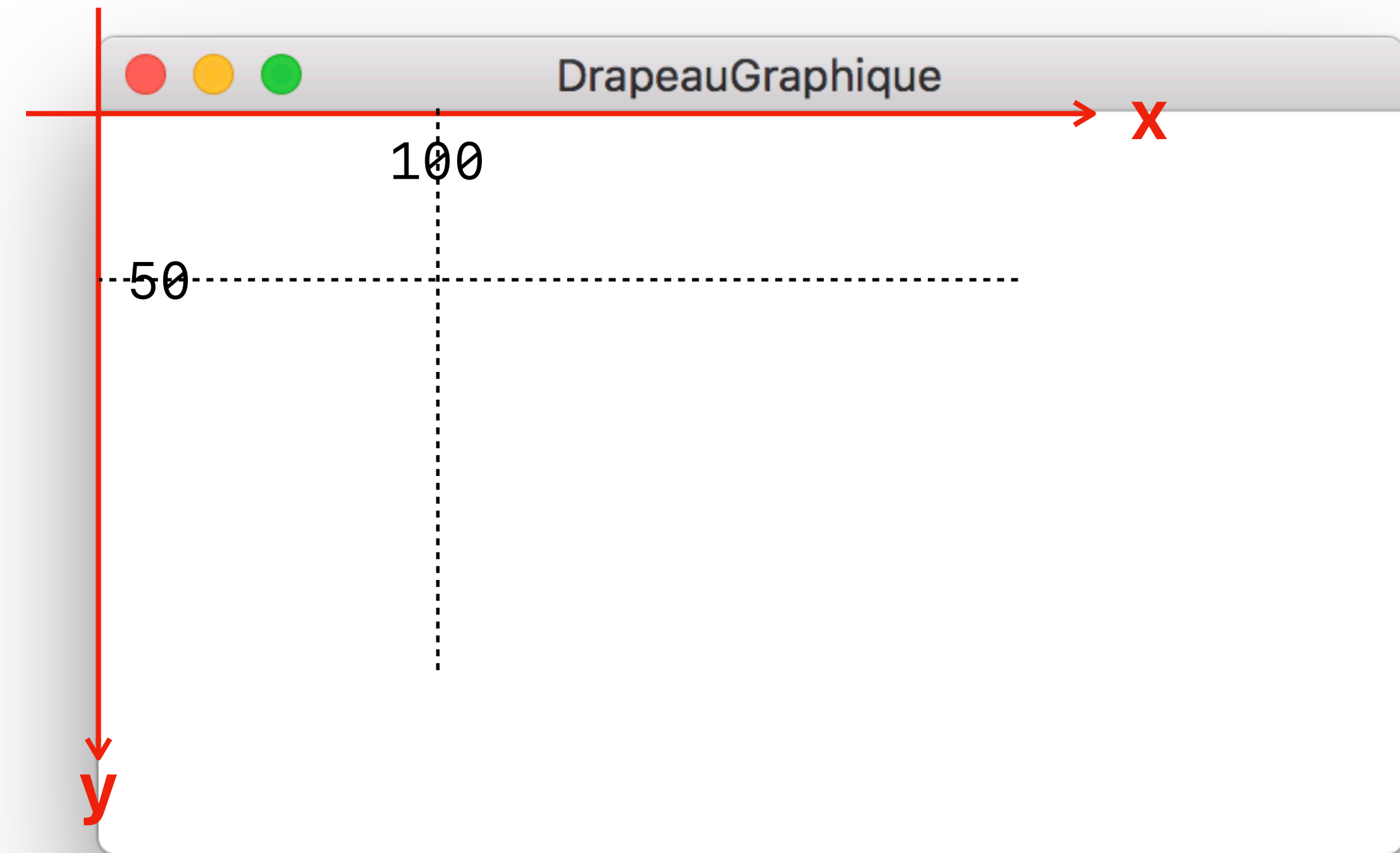
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {
    public void run() {
```



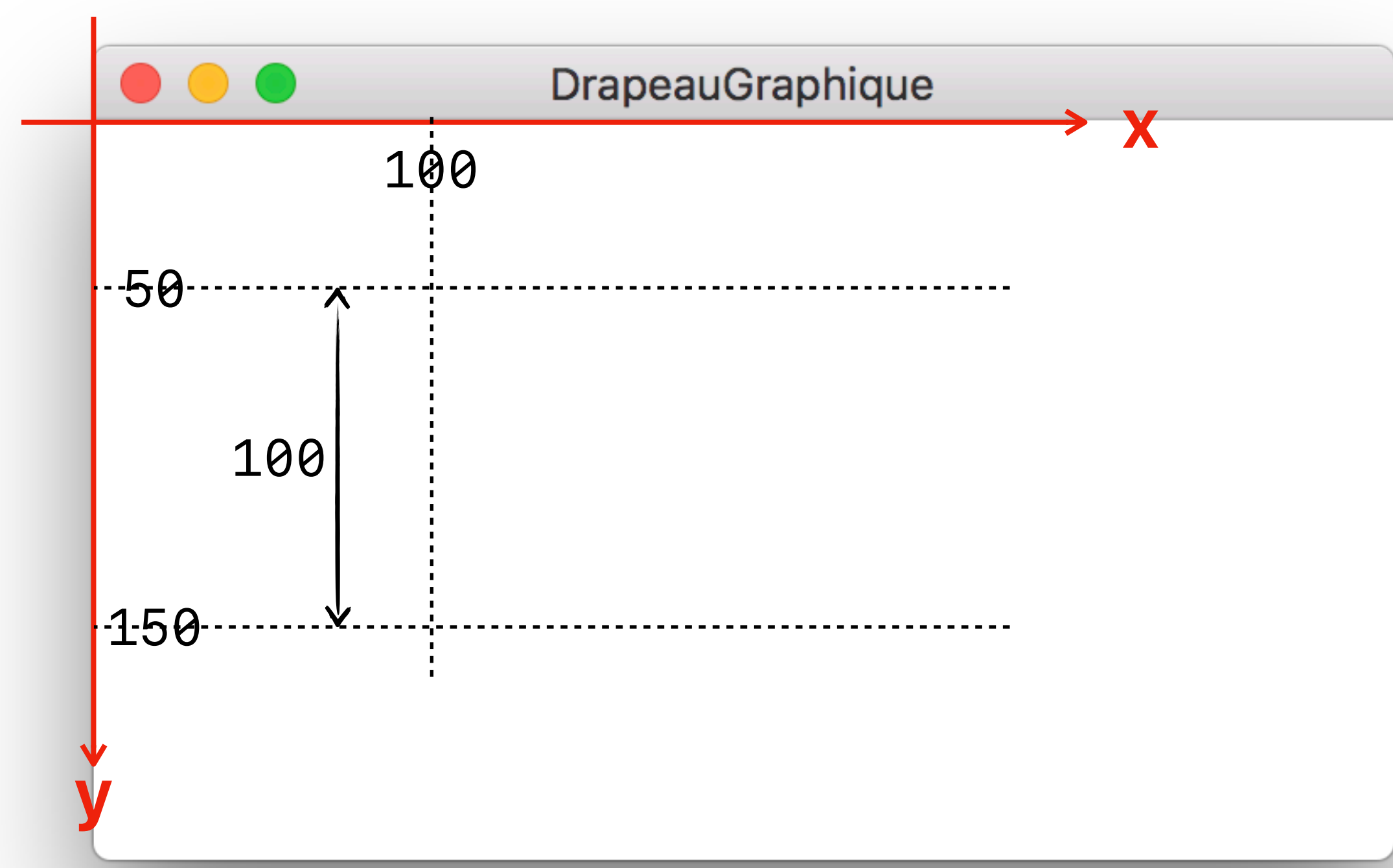
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {
    public void run() {
```



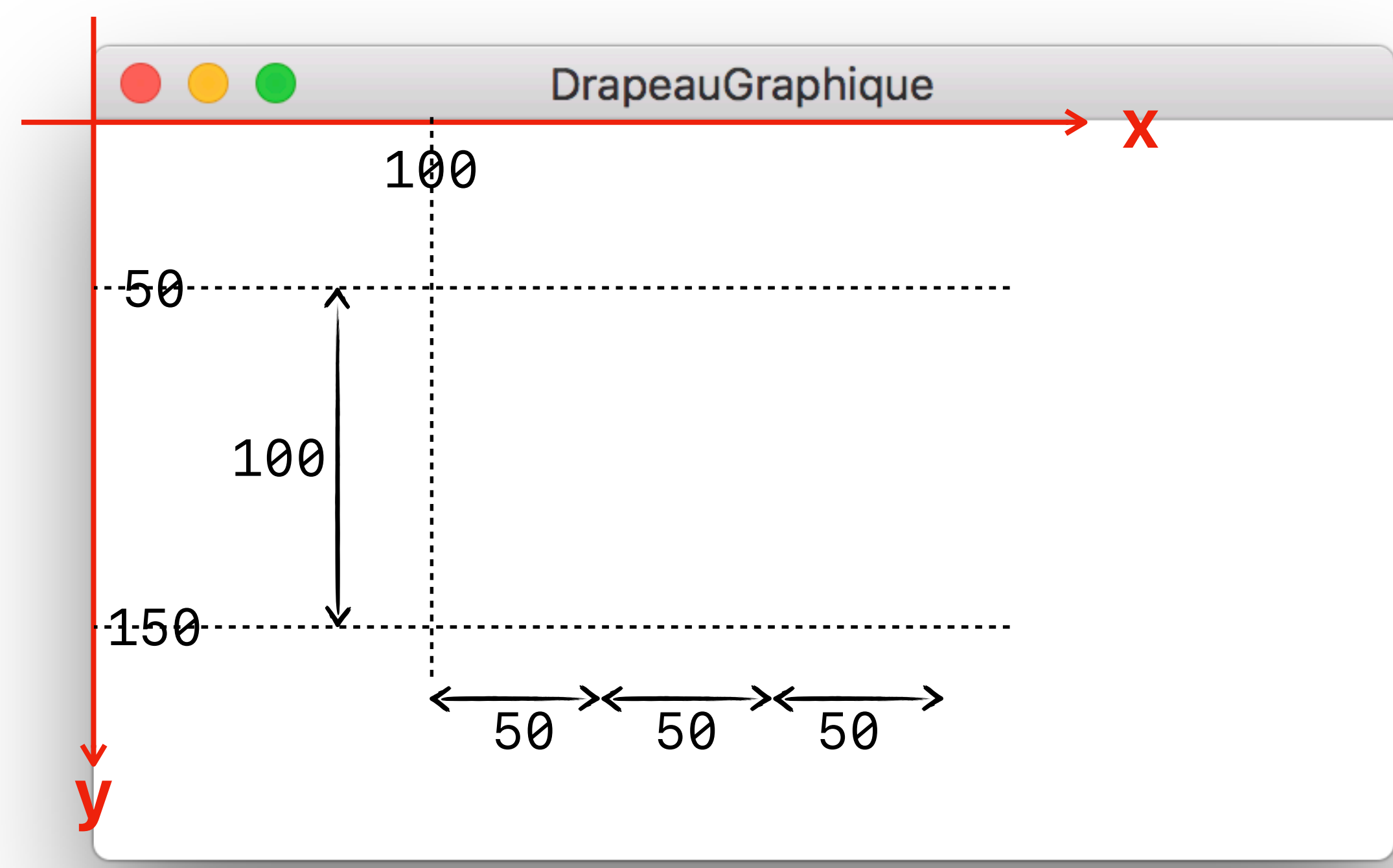
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
  
    }  
}
```



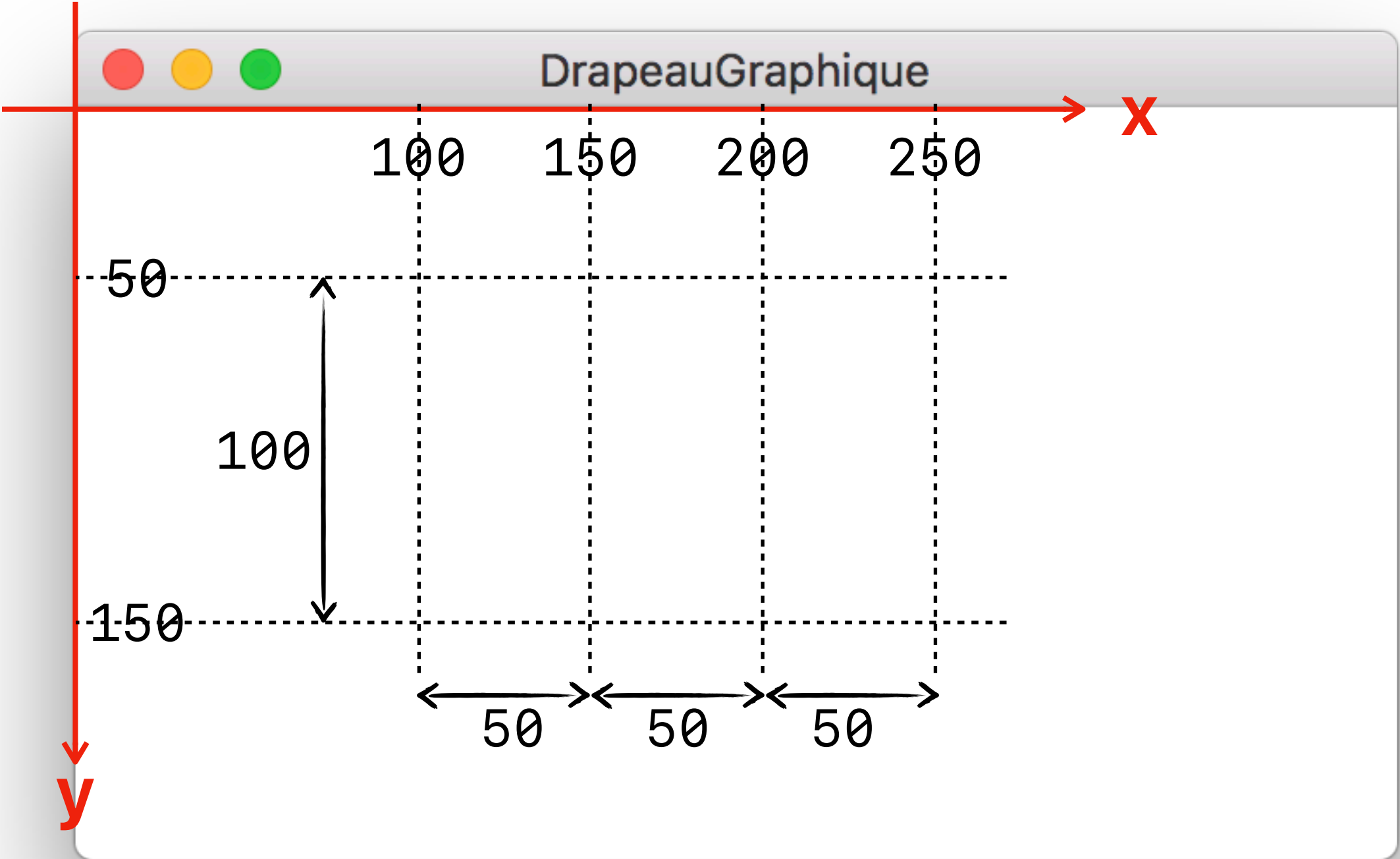
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
  
    }  
}
```



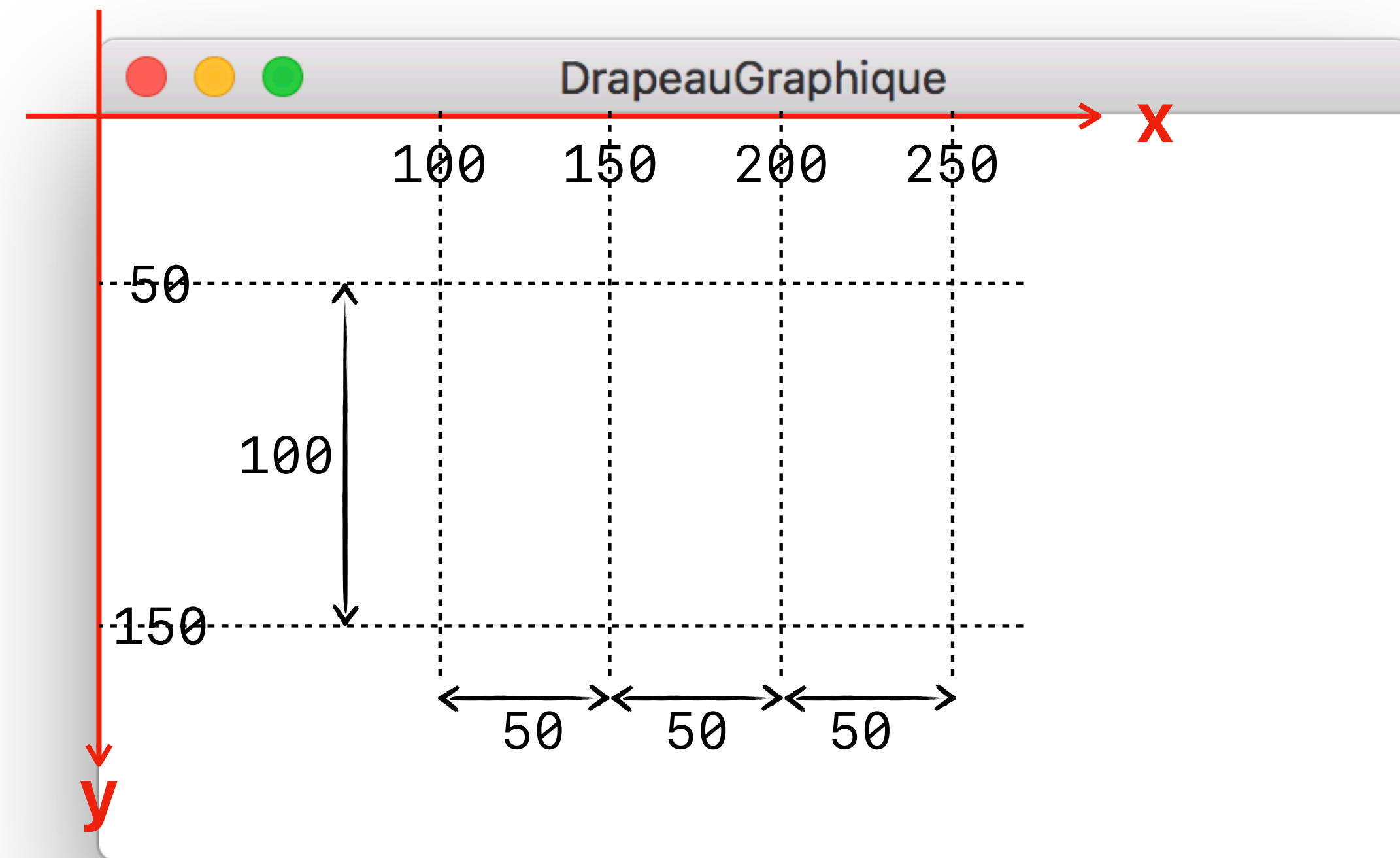
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
  
    }  
}
```



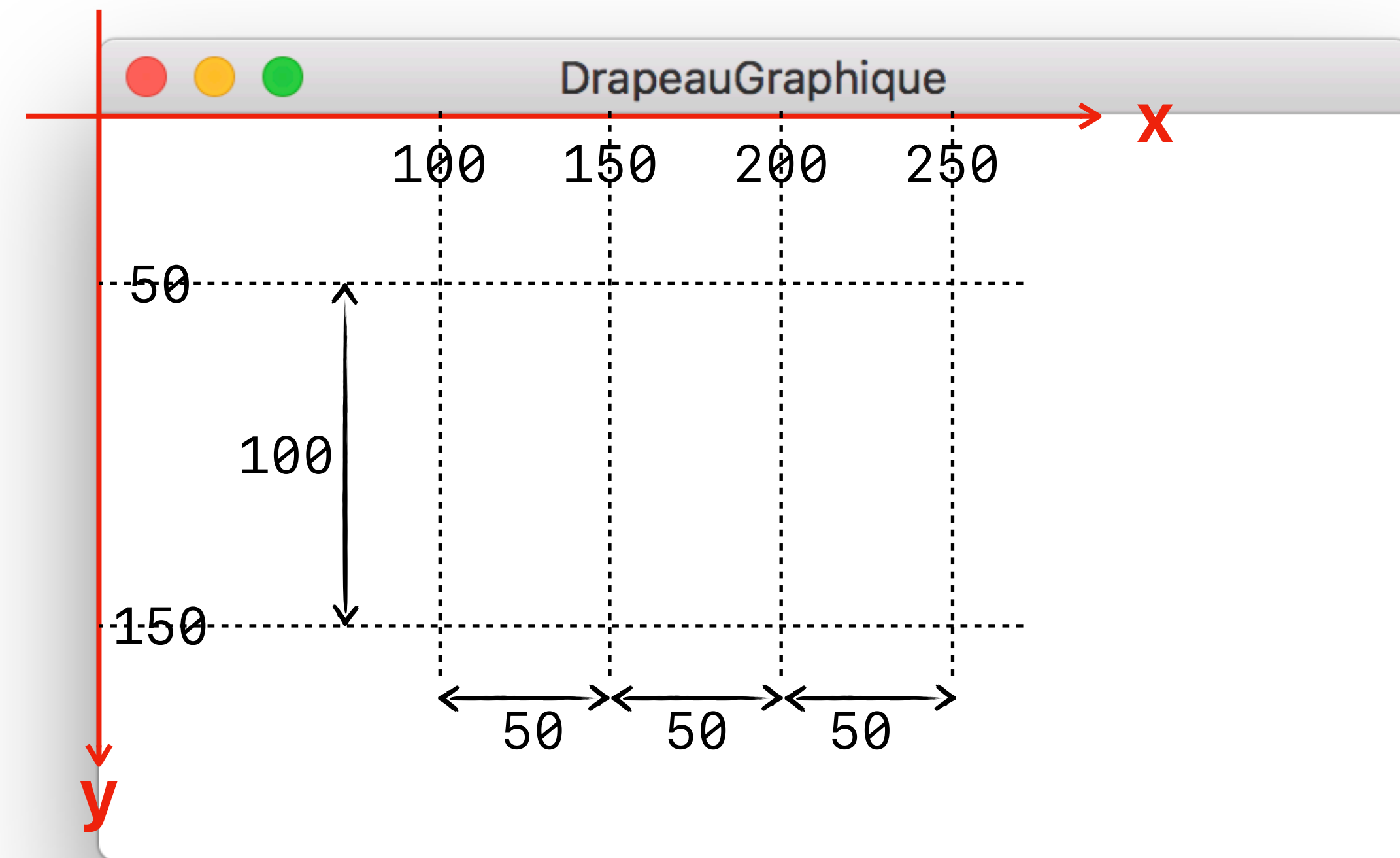
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
    }  
}
```



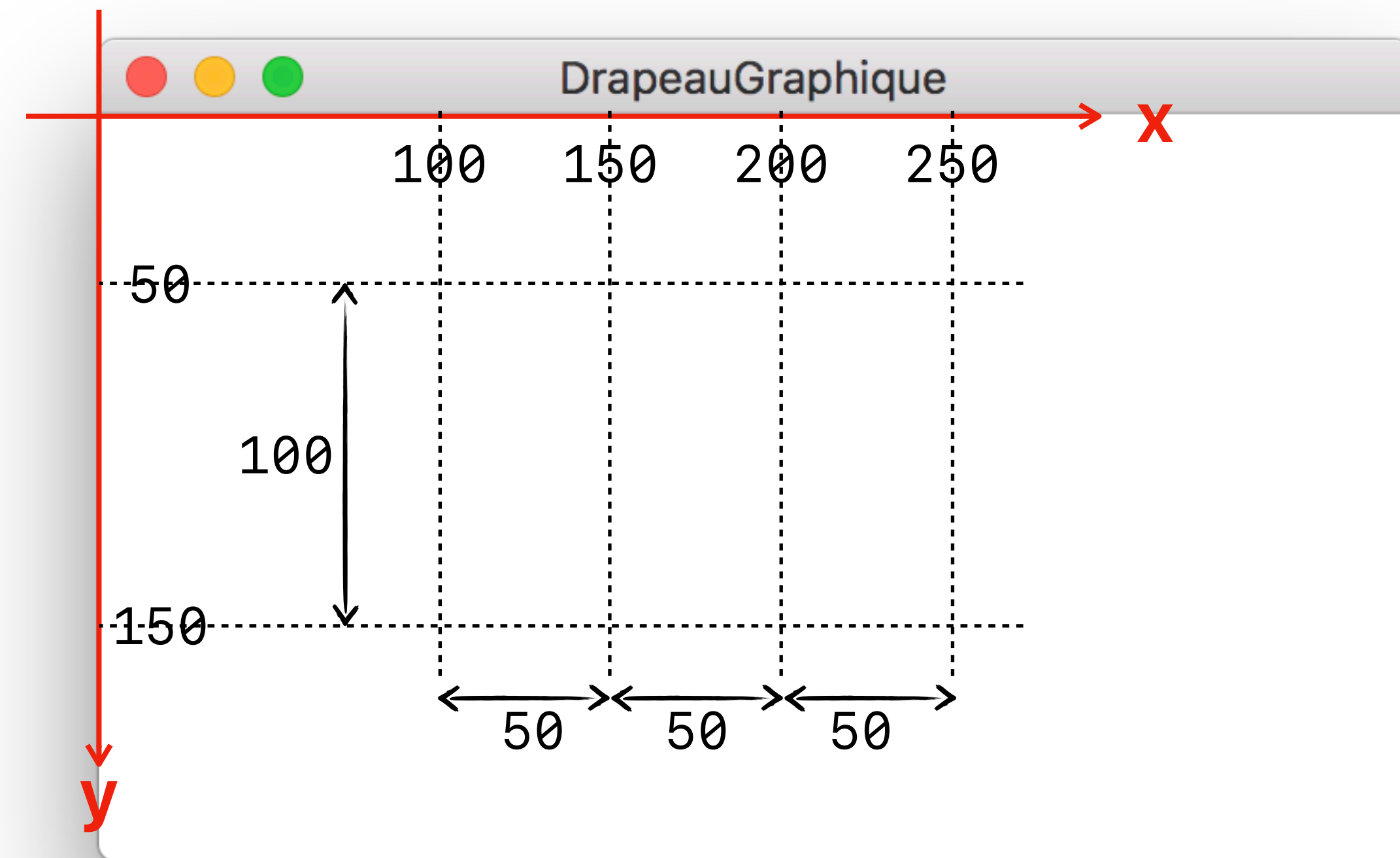
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
    }  
}
```



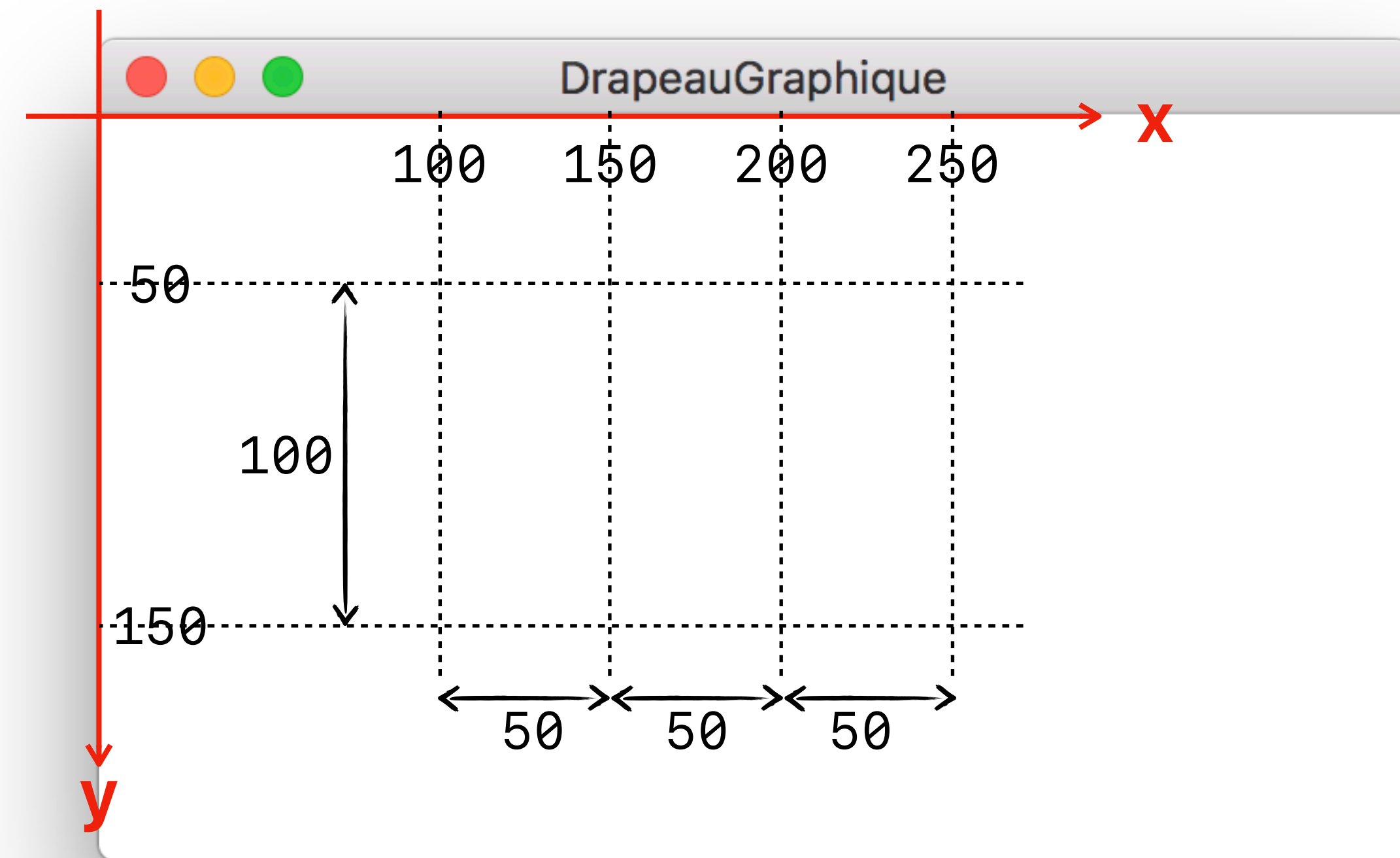
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
    }  
}
```



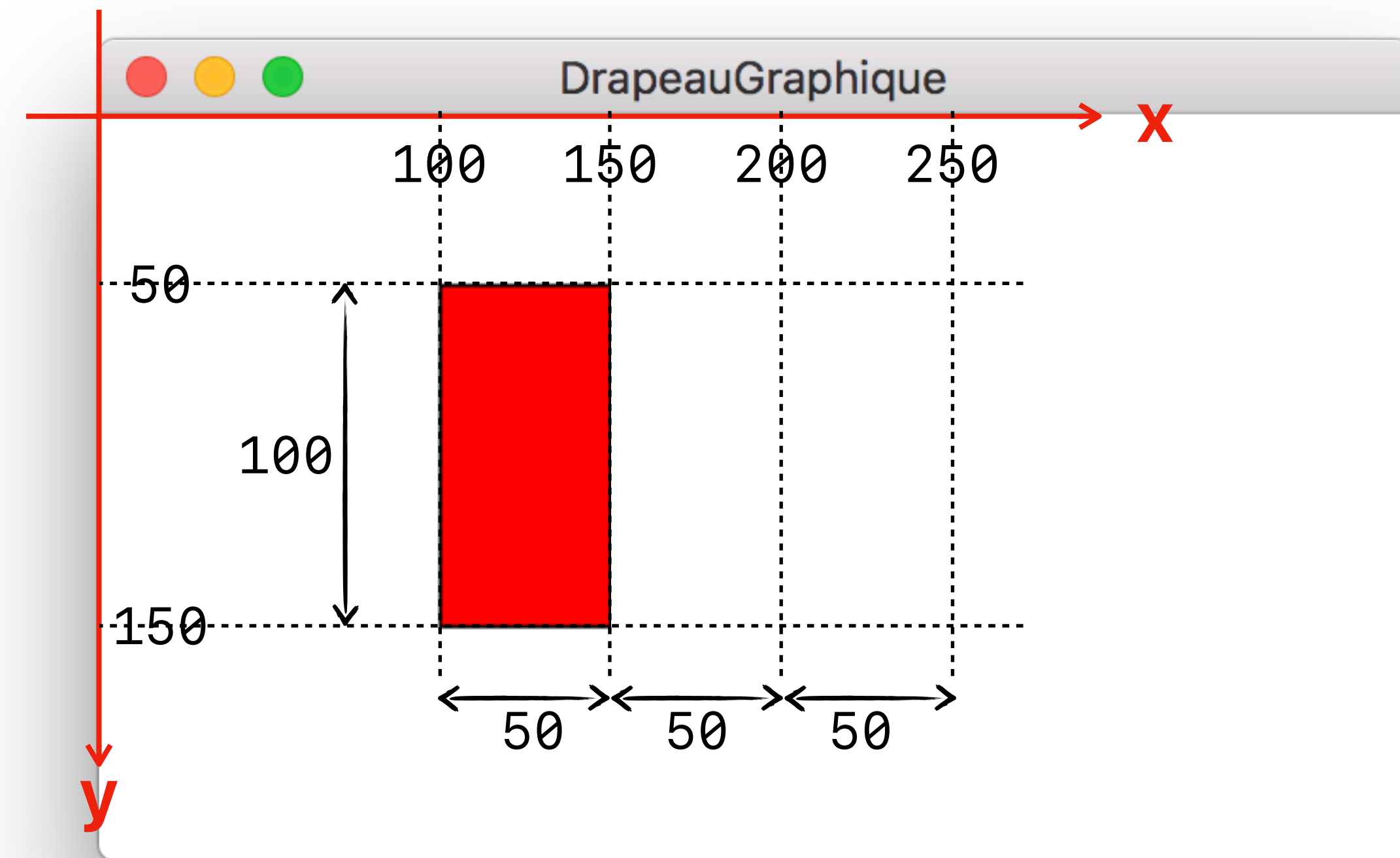
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
    }  
}
```



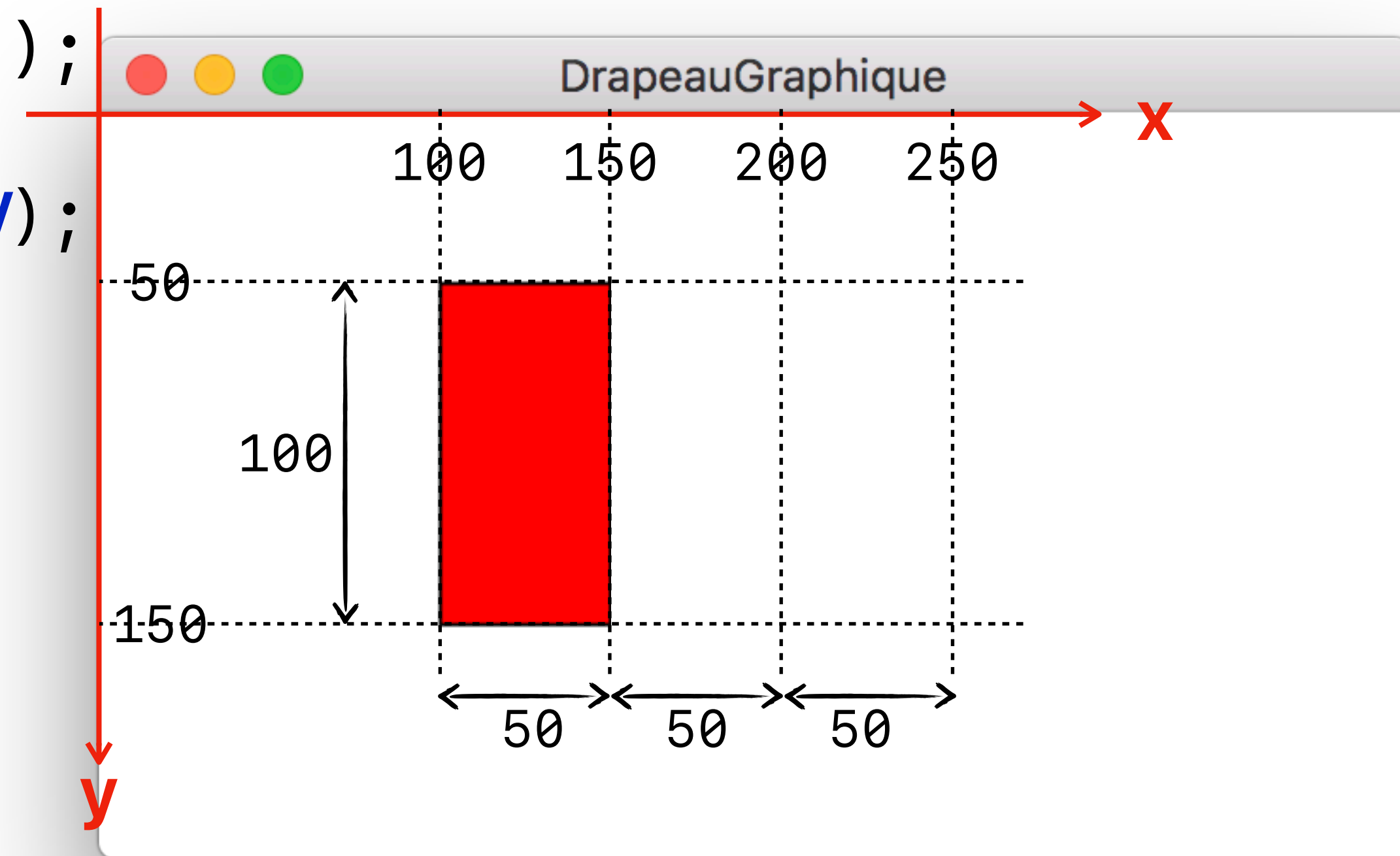
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
    }  
}
```



DrapeauGraphique

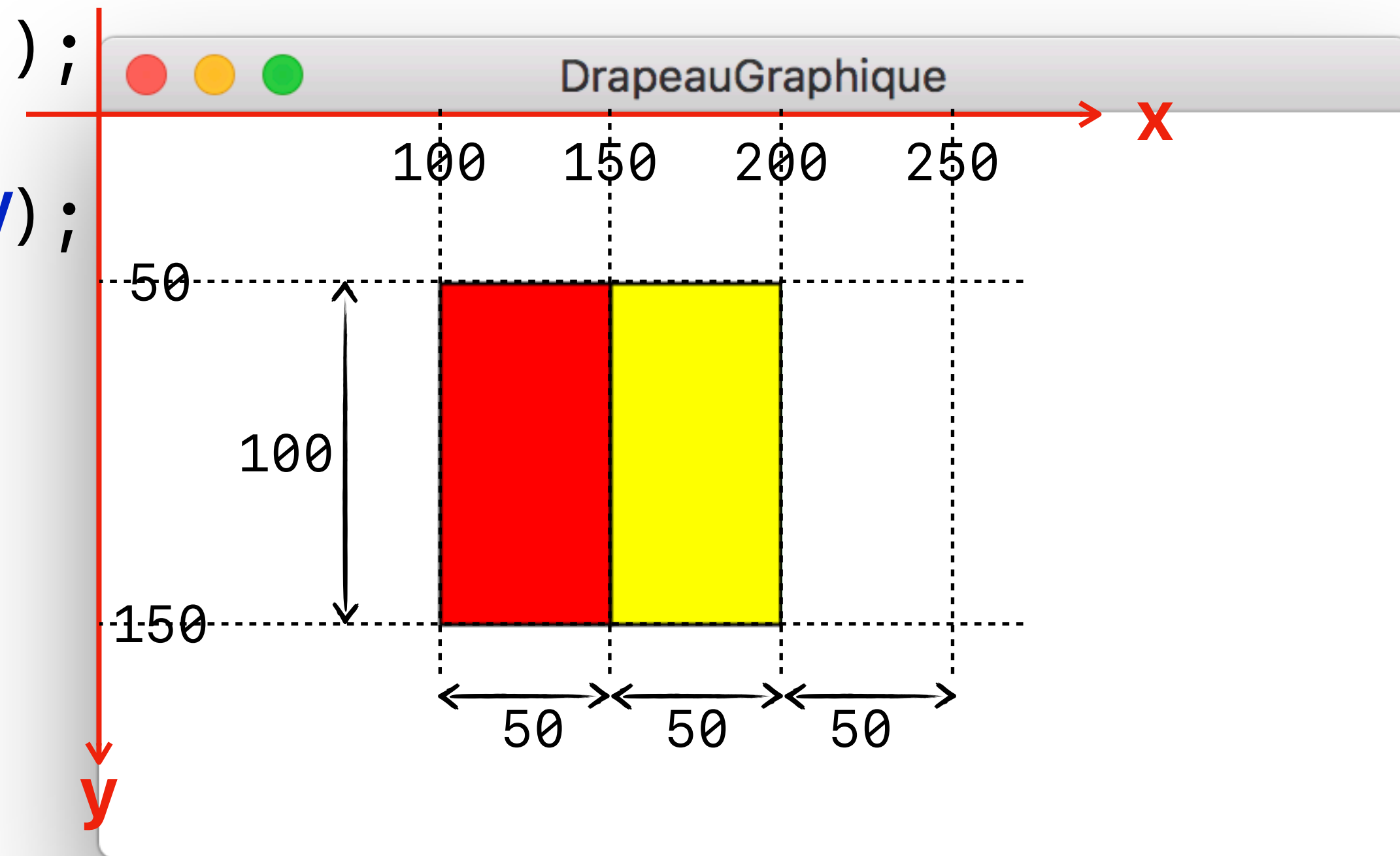
```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
  
        GRect rectJaune = new GRect(50, 100);  
        rectJaune.setFilled(true);  
        rectJaune.setFillColor(Color.YELLOW);  
        add(rectJaune, 150, 50);  
    }  
}
```



DrapeauGraphique

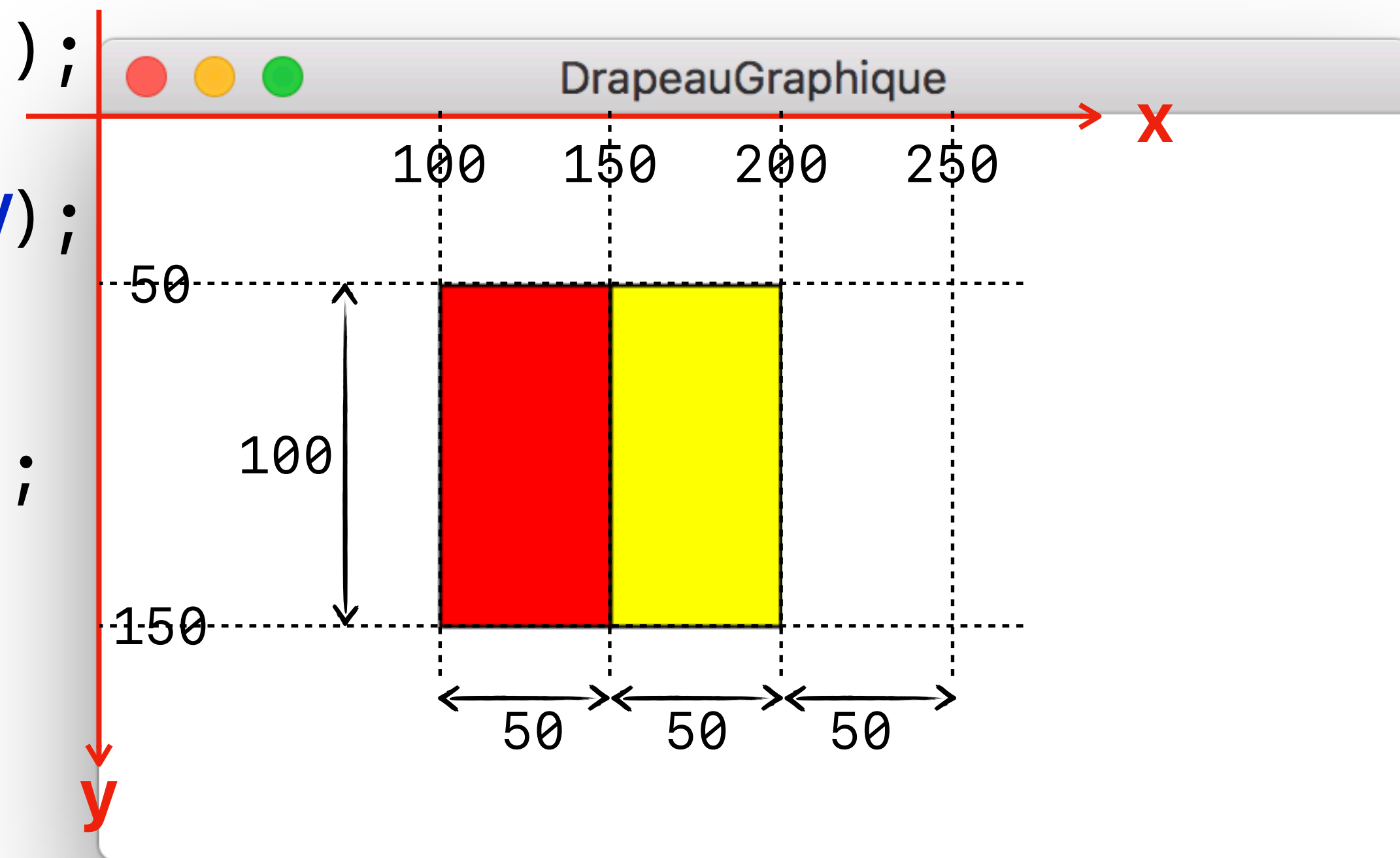
```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);
```

```
        GRect rectJaune = new GRect(50, 100);  
        rectJaune.setFilled(true);  
        rectJaune.setFillColor(Color.YELLOW);  
        add(rectJaune, 150, 50);  
    }  
}
```



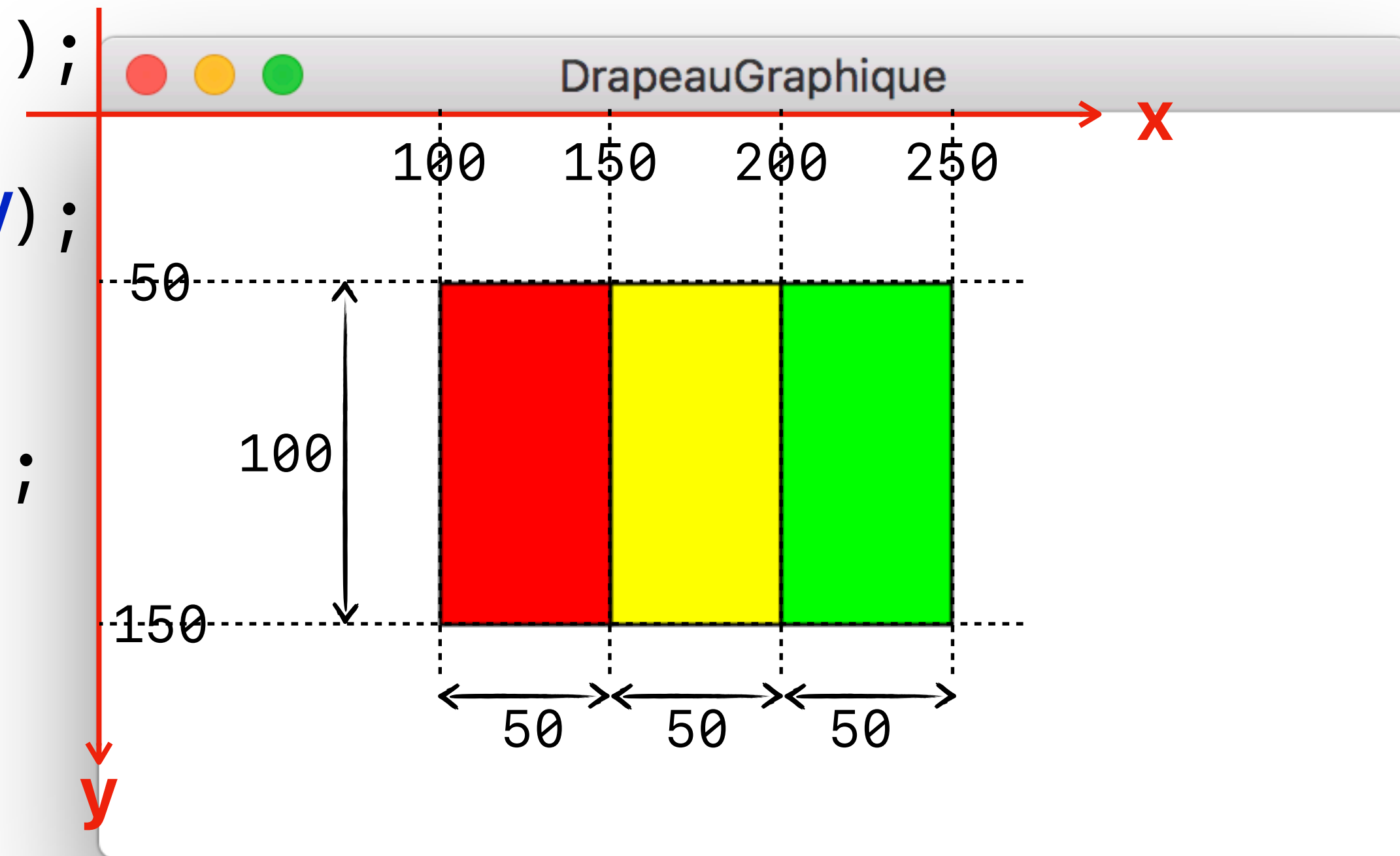
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
  
        GRect rectJaune = new GRect(50, 100);  
        rectJaune.setFilled(true);  
        rectJaune.setFillColor(Color.YELLOW);  
        add(rectJaune, 150, 50);  
  
        GRect rectVert = new GRect(50, 100);  
        rectVert.setFilled(true);  
        rectVert.setFillColor(Color.GREEN);  
        add(rectVert, 200, 50);  
    }  
}
```



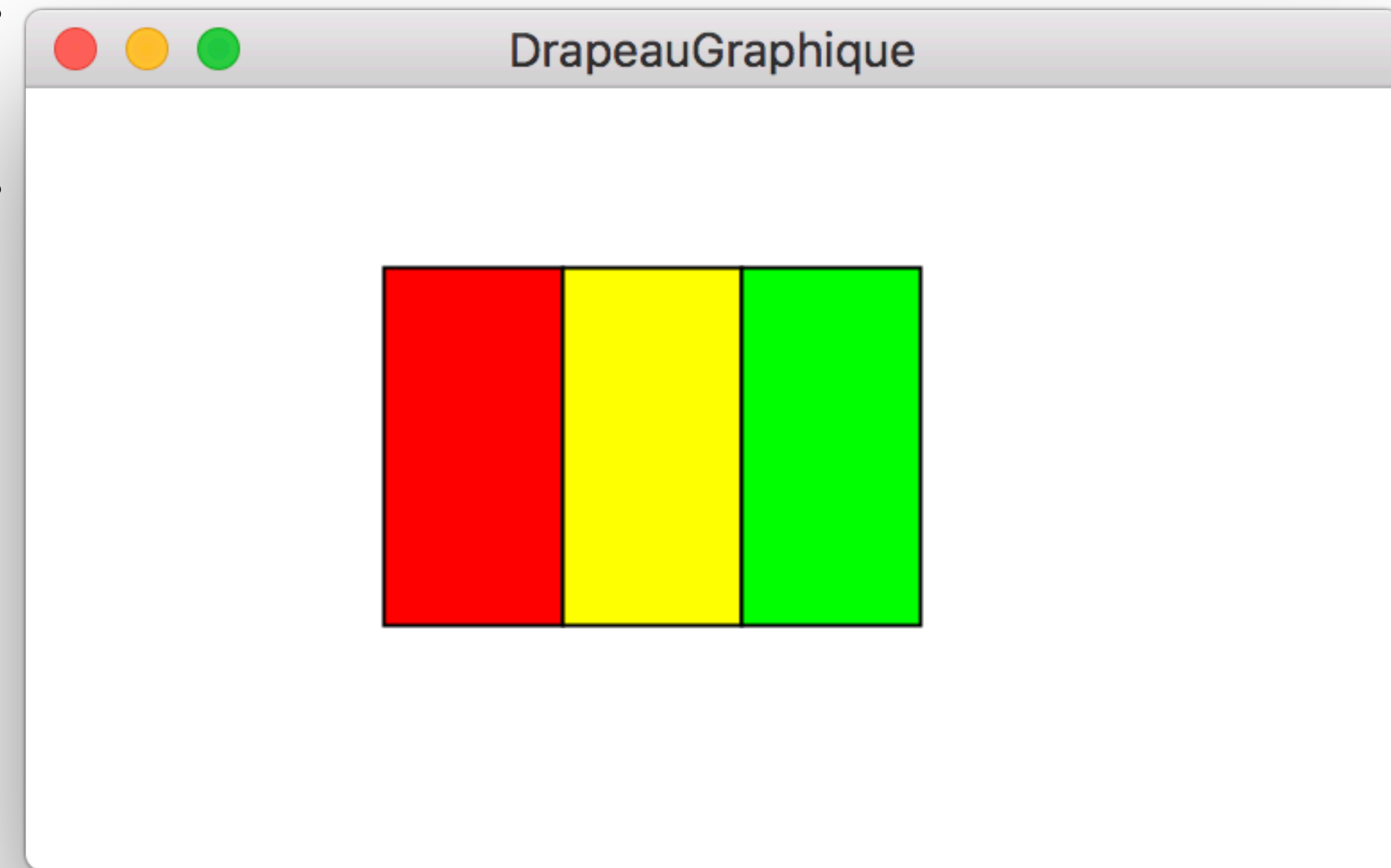
DrapeauGraphique

```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
  
        GRect rectJaune = new GRect(50, 100);  
        rectJaune.setFilled(true);  
        rectJaune.setFillColor(Color.YELLOW);  
        add(rectJaune, 150, 50);  
  
        GRect rectVert = new GRect(50, 100);  
        rectVert.setFilled(true);  
        rectVert.setFillColor(Color.GREEN);  
        add(rectVert, 200, 50);  
    }  
}
```



DrapeauGraphique

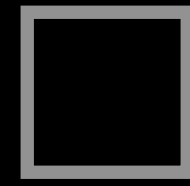
```
public class DrapeauGraphique extends GraphicsProgram {  
    public void run() {  
        GRect rectRouge = new GRect(50, 100);  
        rectRouge.setFilled(true);  
        rectRouge.setFillColor(Color.RED);  
        add(rectRouge, 100, 50);  
  
        GRect rectJaune = new GRect(50, 100);  
        rectJaune.setFilled(true);  
        rectJaune.setFillColor(Color.YELLOW);  
        add(rectJaune, 150, 50);  
  
        GRect rectVert = new GRect(50, 100);  
        rectVert.setFilled(true);  
        rectVert.setFillColor(Color.GREEN);  
        add(rectVert, 200, 50);  
    }  
}
```



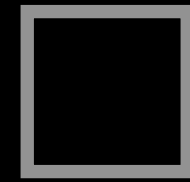
Objectif de ce cours



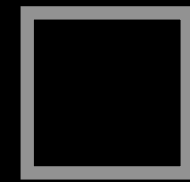
Variables, révision



GraphicsProgram: Introduction et exemple

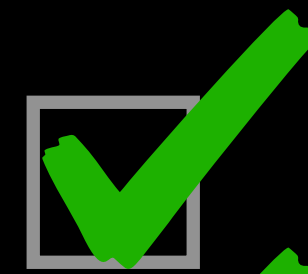


Quelques objets graphiques communs



Grand-père de Karel

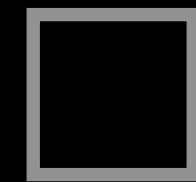
Objectif de ce cours



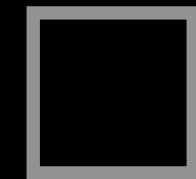
Variables, révision



GraphicsProgram: Introduction et exemple



Quelques objets graphiques communs



Grand-père de Karel

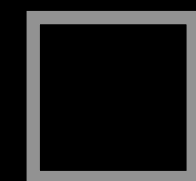
Objectif de ce cours



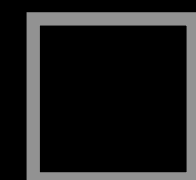
Variables, révision



GraphicsProgram: Introduction et exemple

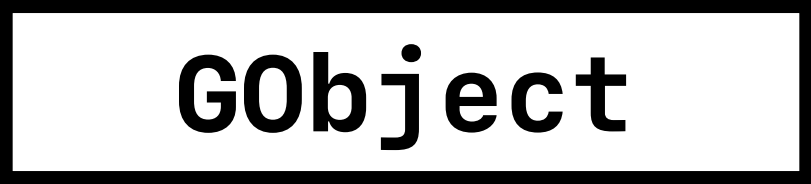


Quelques objets graphiques communs

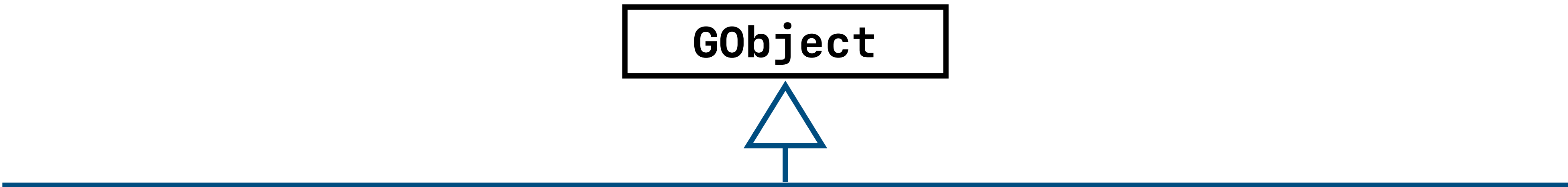


Grand-père de Karel

Hierarchie des objets graphiques

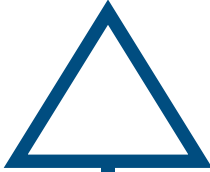


Hierarchie des objets graphiques

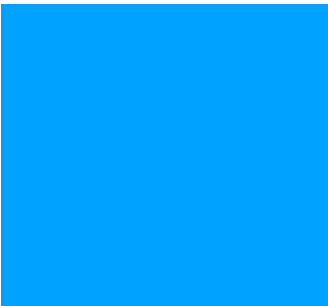


Hierarchie des objets graphiques

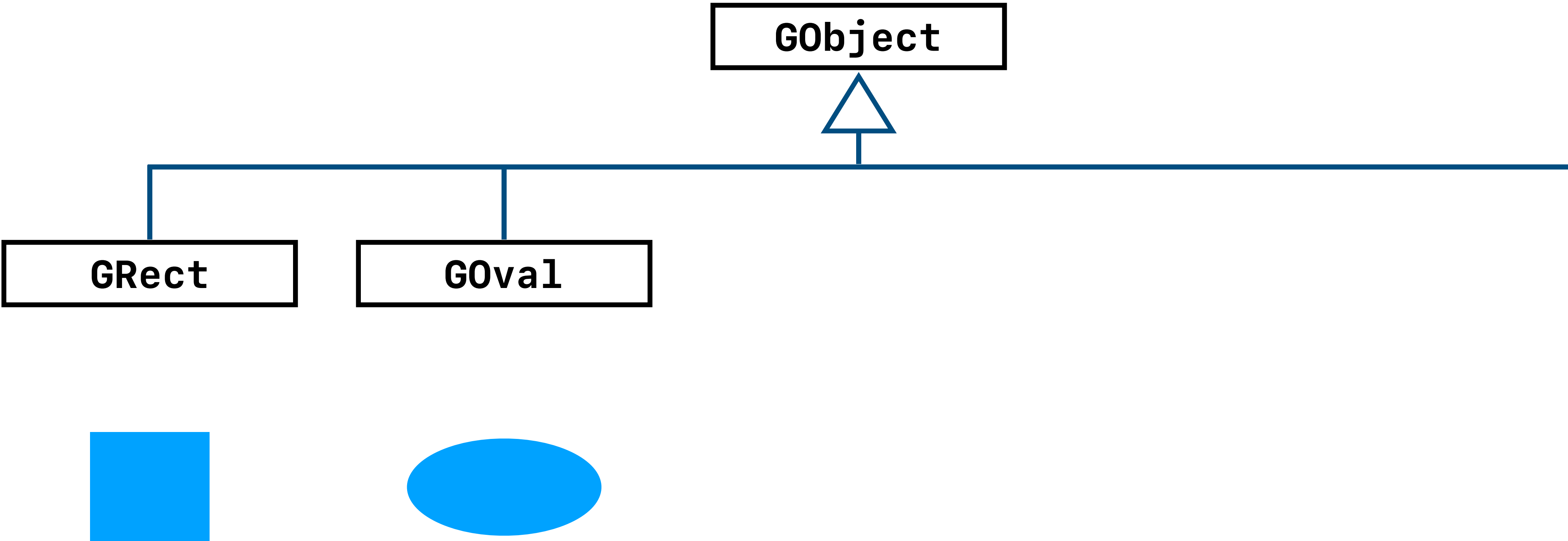
GObject



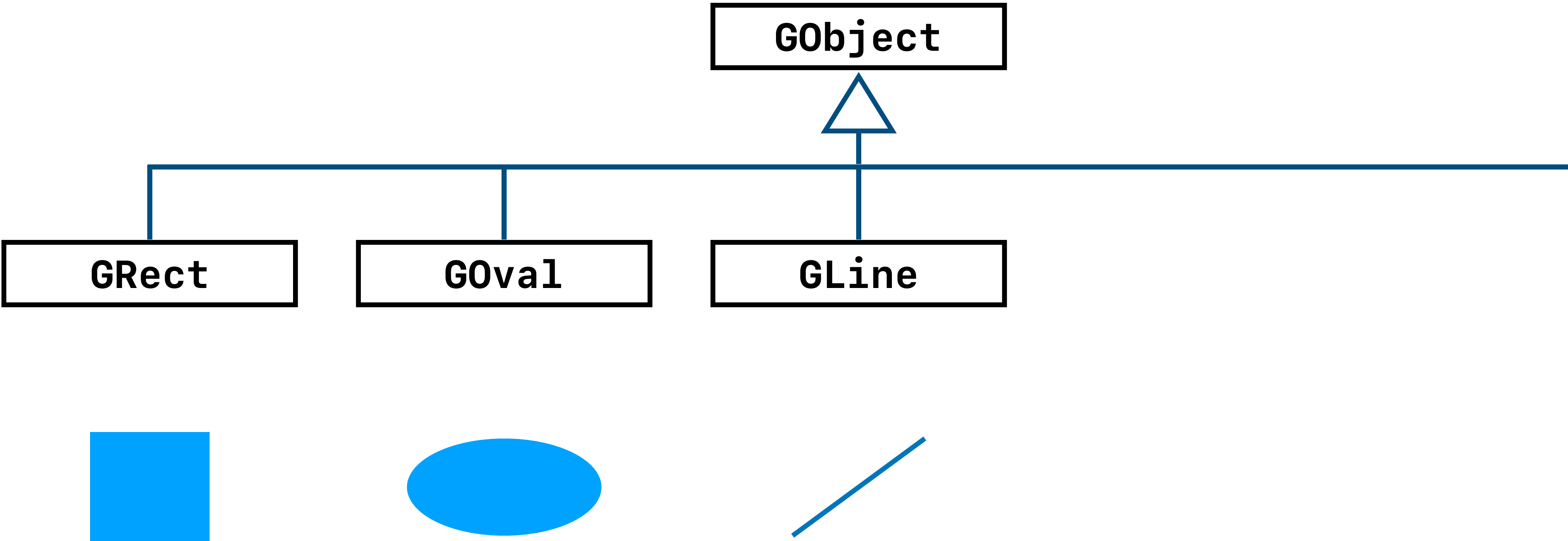
GRect



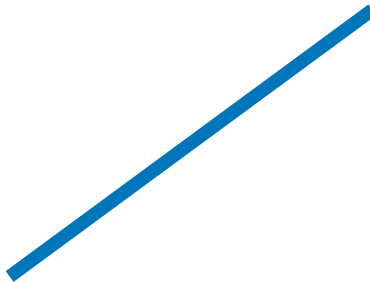
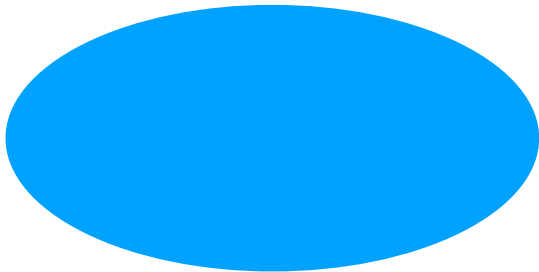
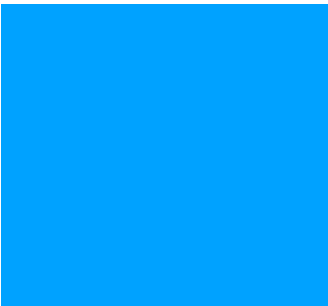
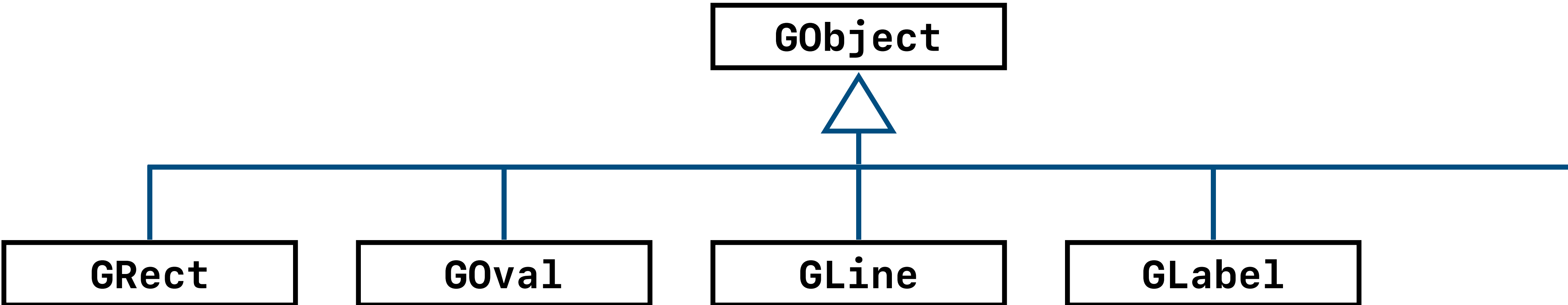
Hierarchie des objets graphiques



Hierarchie des objets graphiques

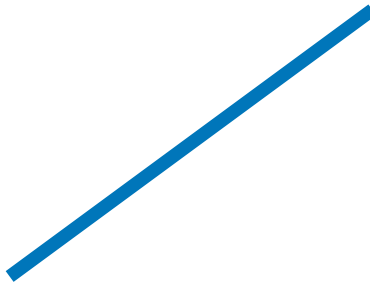
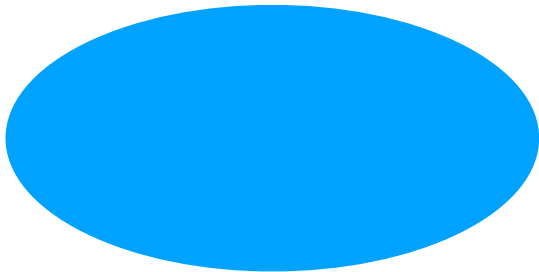
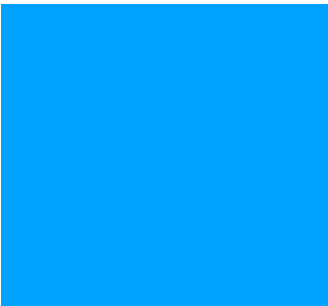
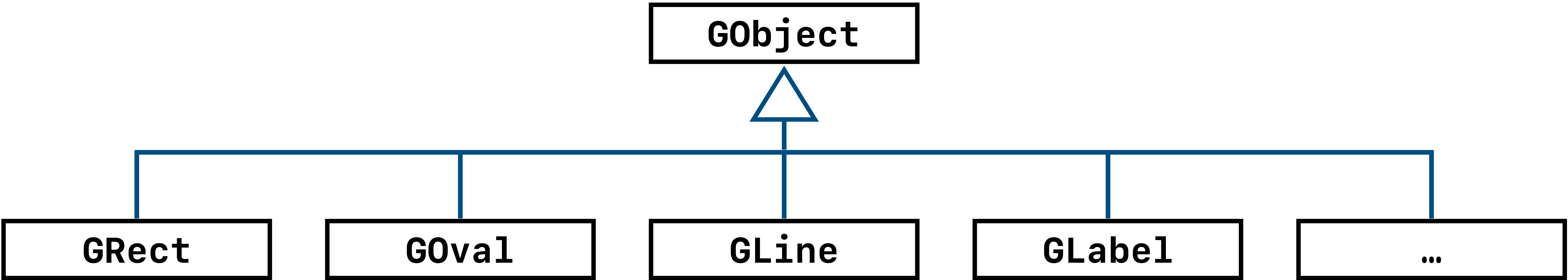


Hierarchie des objets graphiques



Grand-père de Karel

Hierarchie des objets graphiques



Grand-père de Karel

Types primitifs vs Objets

Types primitifs vs Objets

char

double

int

boolean

Types primitifs vs Objets

char

double

int

boolean



String

Integer

Long

Float

Boolean

Character

Byte

Short

Void

Object

Class

Interface

Types primitifs vs Objets

char

double

int

boolean

RandomGenerator

GRect

GLine

GOval

GLabel

Types primitifs vs Objets

char

double

int

boolean

```
int nbEleves = 11;  
nbEleves = 10;
```

RandomGenerator

GRect

GLine

G Oval

GLabel

Types primitifs vs Objets

char

double

int

boolean

```
int nbEleves = 11;  
nbEleves = 10;
```

RandomGenerator

GRect

GLine

G Oval

GLabel

```
RandomGenerator rg = new RandomGenerator();  
rg.nextInt();
```


Types primitifs vs Objets

char

double

int

boolean

```
int nbEleves = 11;  
nbEleves = 10;
```

RandomGenerator

GRect

GLine

G Oval

GLabel

```
RandomGenerator rg = new RandomGenerator();  
rg.nextInt();
```

```
GRect rect = new GRect(80, 200);  
rect.setLocation(50, 100);  
rect.setFilled(true);  
rect.setFillColor(Color.RED);
```

Méthodes

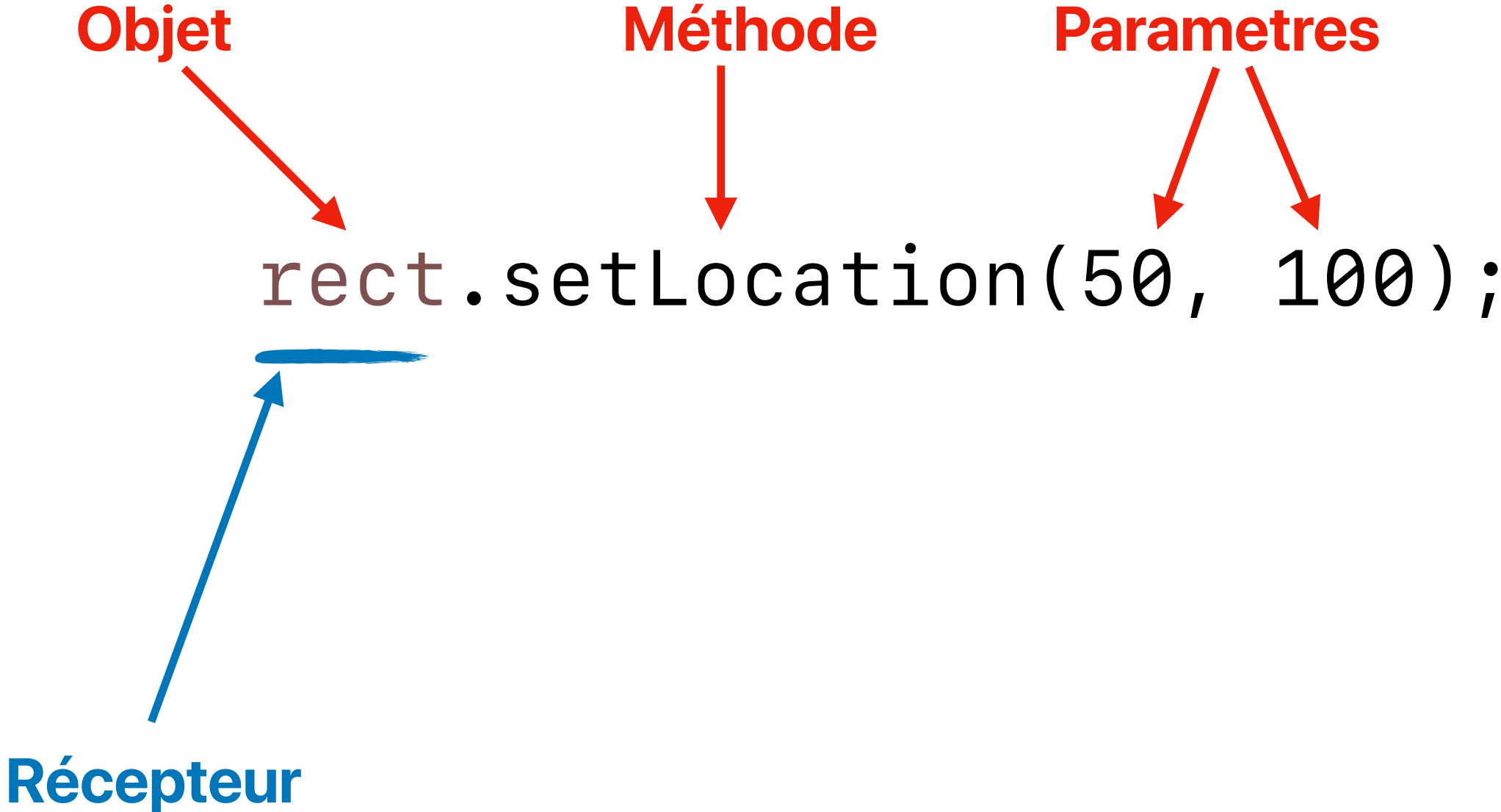
Objet

Méthode

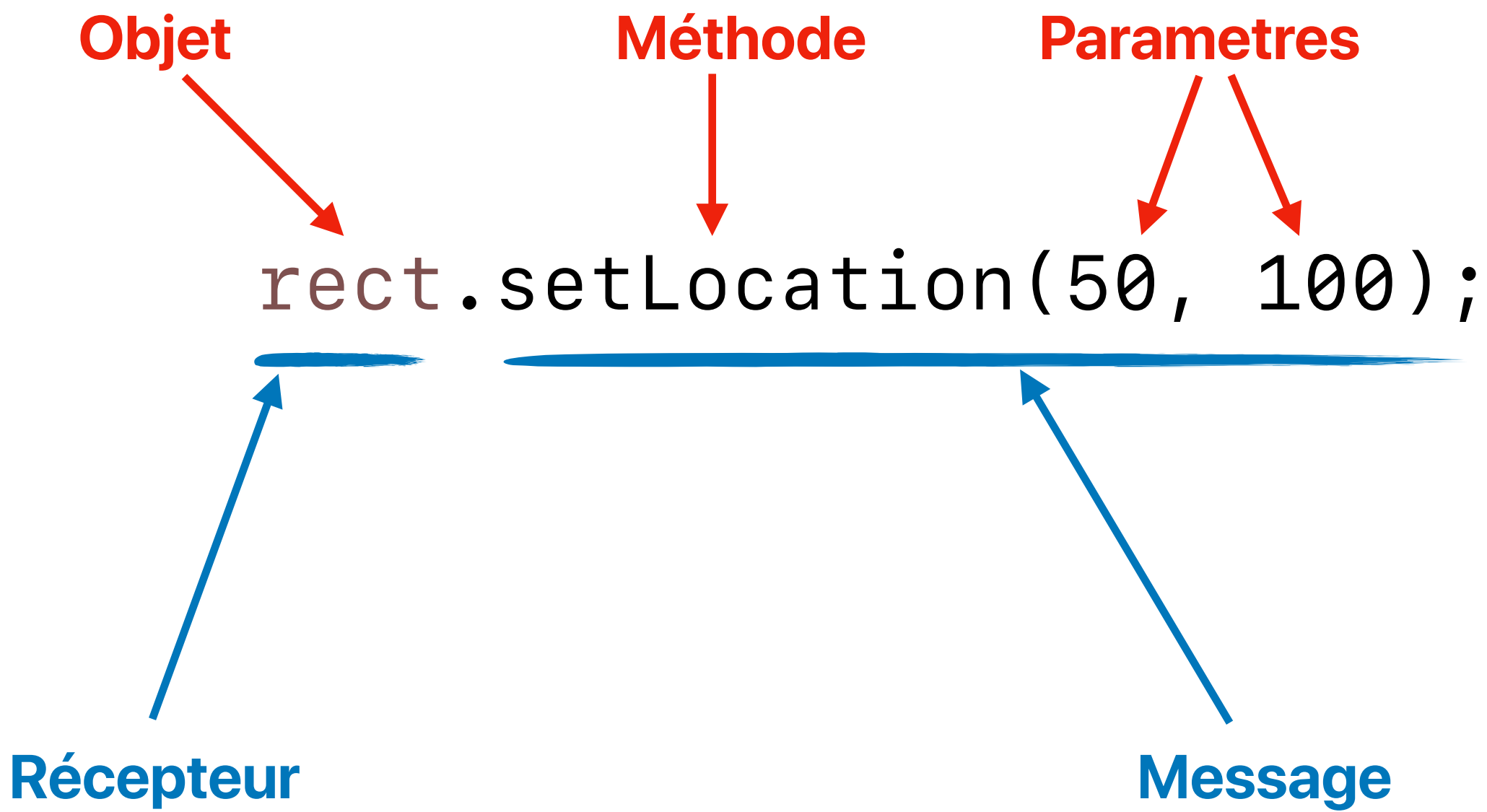
Parametres

rect.setLocation(50, 100);

Méthodes



Méthodes



Methodes de GObject

Ces méthodes sont héritées par tout les objets graphiques

void setLocation(double x, double y) Met à jour la position d'un objet pour correspondre aux coordonnées (x, y) spécifiées.
void move(double dx, double dy) Déplace un objet de sa position courante par un déplacement de (dx, dy).
double getWidth() Retourne la largeur de l'objet.
double getHeight() Retourne la hauteur de l'objet.
void setColor(Color c) Met à jour la couleur de l'objet pour être la couleur spécifiée , c.

GRect

```
GRect rect1 = new GRect(largeur, hauteur);
```

```
GRect rect2 = new GRect(x, y, largeur, hauteur);
```

GOval

```
GOval oval1 = new GOval(largeur, hauteur);
```

```
GOval oval2 = new GOval(x, y, largeur, hauteur);
```

```
GOval circle1 = new GOval(x, y, diametre, diametre);
```

Methodes de spécifiques à GRect et GOval

void setFilled(boolean fill) Spécifie si l'objet est rempli (true) ou non (false).
boolean isFilled() Retourne true si l'objet est rempli; retourne false dans le cas contraire.
void setFillColor(Color c) Spécifie la couleur de remplissage de l'objet. Si c est null , la couleur de l'objet est utilisée pour remplir.

GLine

```
GLine l1 = new GLine(x1, x2, y1, y2);
```

GLabel

```
GLabel label1 = new GLabel("Grand-père de Karel");
```

```
GLabel label2 = new GLabel("Grand-père de Karel", x, y);
```

Méthodes Spécifiques à GLabel

void setFont(String fontName) Spécifie la fonte du texte. Voir chapitre 5 pour plus d'information.
double getAscent() Retourne la hauteur du texte a en dessus de la ligne de de base.
double getDescent() Retourne la hauteur du texte a en dessous de la ligne de base.



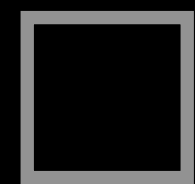
Objectif de ce cours



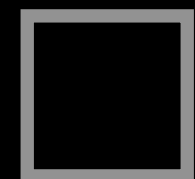
Variables, révision



GraphicsProgram: Introduction et exemple



Quelques objets graphiques communs



Grand-père de Karel

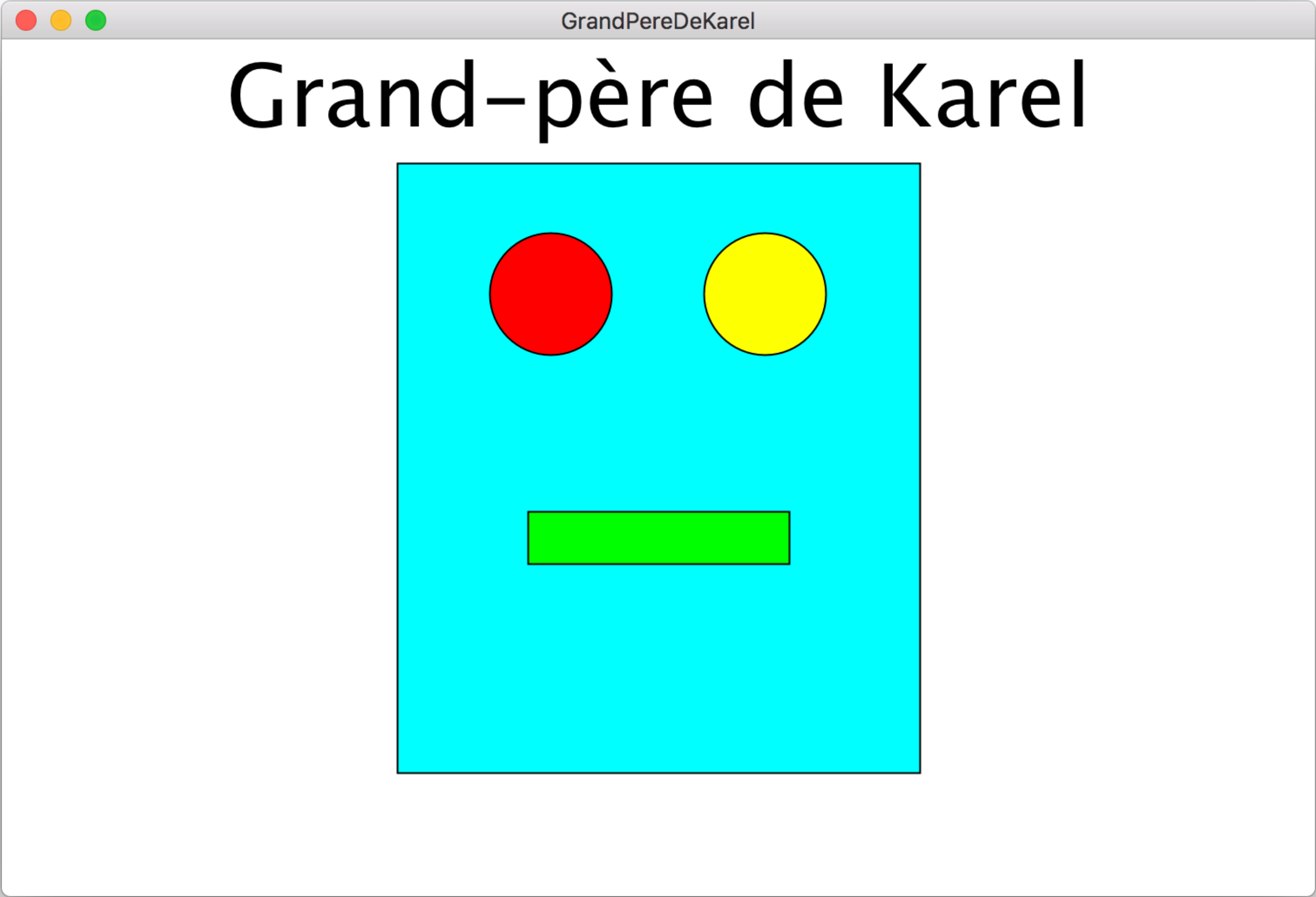
Objectif de ce cours

- ☒ Variables, révision
- ☒ GraphicsProgram: Introduction et exemple
- ☒ Quelques objets graphiques communs
- ☐ Grand-père de Karel

Objectif de ce cours

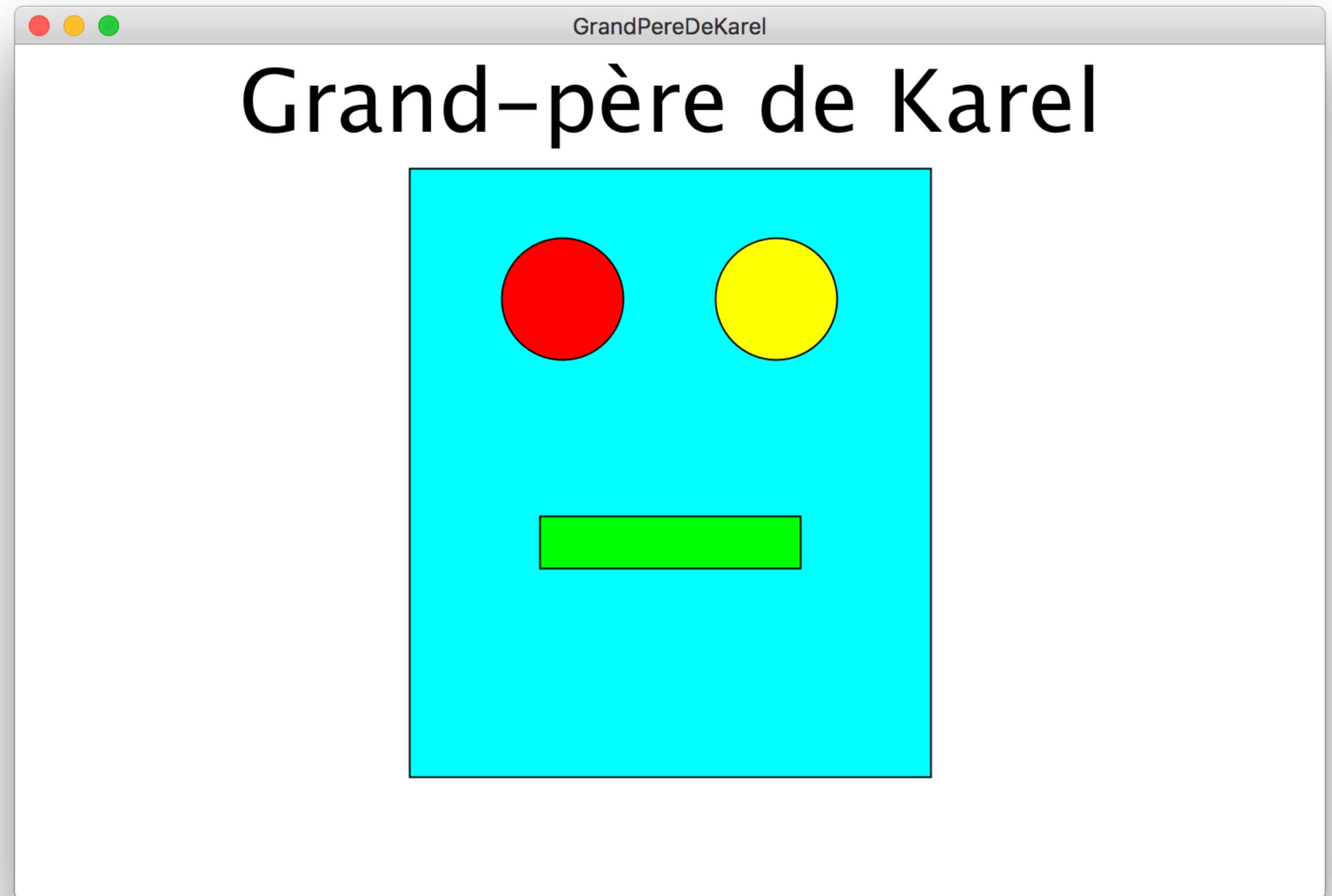
- ☒ Variables, révision
- ☒ GraphicsProgram: Introduction et exemple
- ☒ Quelques objets graphiques communs
- ☐ Grand-père de Karel

Grand-père de Karel



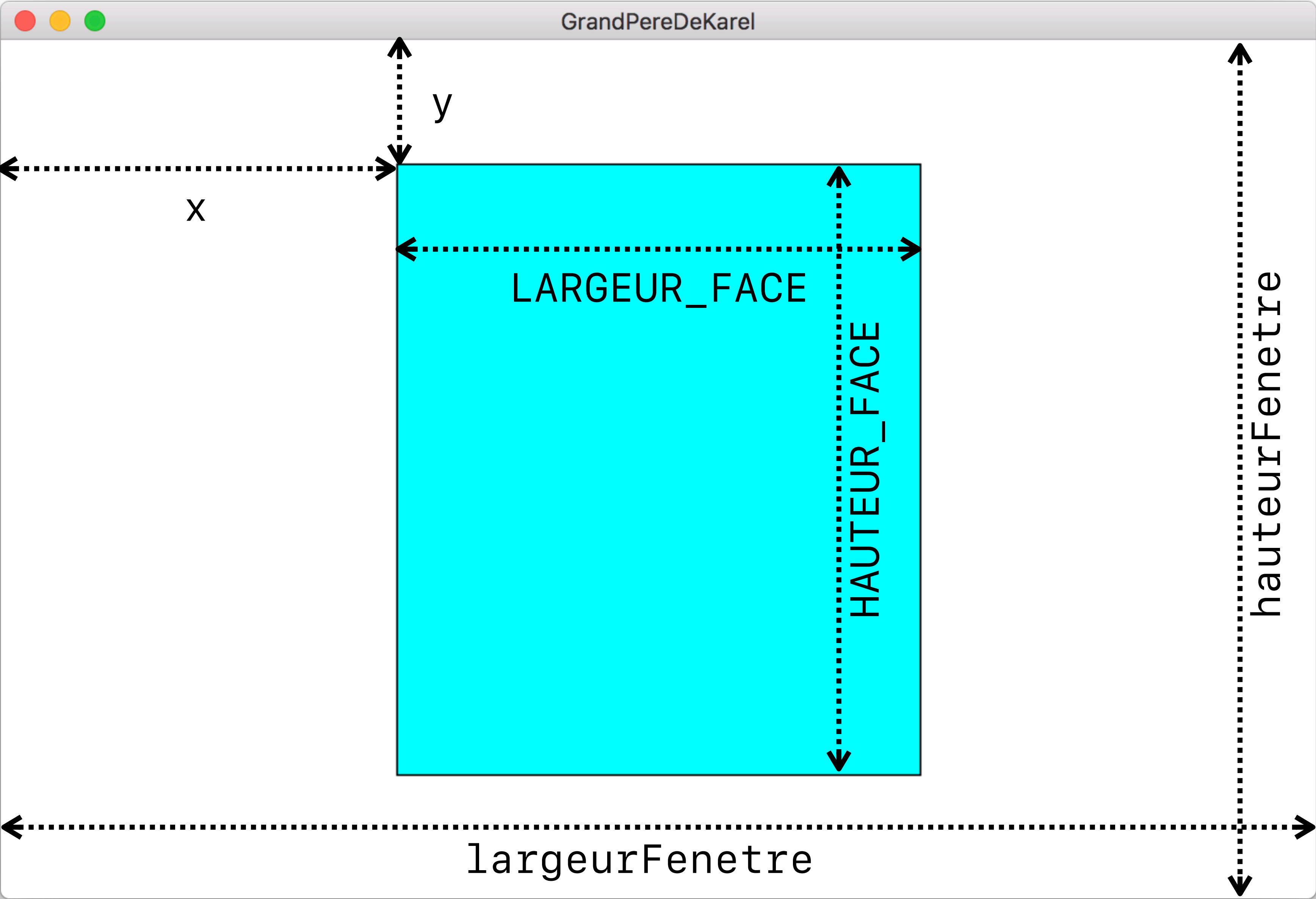
Grand-père de Karel

```
public void run() {  
    dessinerTete();  
    dessinerOeilGauche();  
    dessinerOeilDroit();  
    dessinerBouche();  
    dessinerEtiquette();  
}
```



Grand-père de Karel

```
private void  
double  
double  
  
double  
double  
  
GRect  
tete.s  
tete.s  
  
add(tete)  
  
}
```

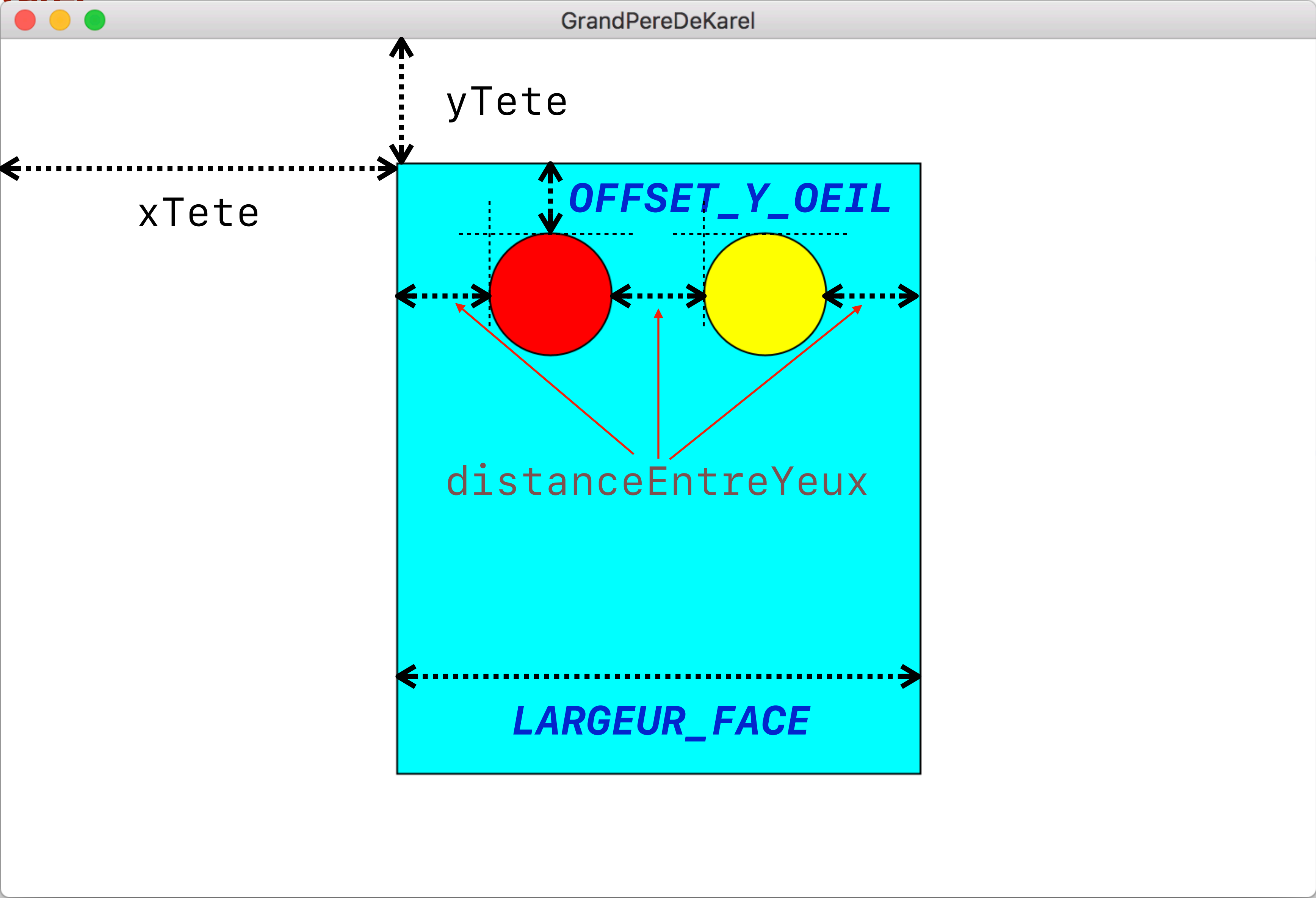


Grand-père de Karel

```
private void dessinerTete() {  
    double largeurFenetre = getWidth();  
    double hauteurFenere = getHeight();  
  
    double x = (largeurFenetre - LARGEUR_FACE) / 2;  
    double y = (hauteurFenere - HAUTEUR_FACE) / 2;  
  
    GRect tete = new GRect(x, y, LARGEUR_FACE, HAUTEUR_FACE);  
    tete.setFilled(true);  
    tete.setFillColor(Color.CYAN);  
  
    add(tete);  
}
```

Grand-père de Karel

```
private void  
double  
double  
double  
  
double  
double  
GOval  
oeil.s  
oeil.s  
add(oe  
  
}
```



)/3;

);

Grand-père de Karel

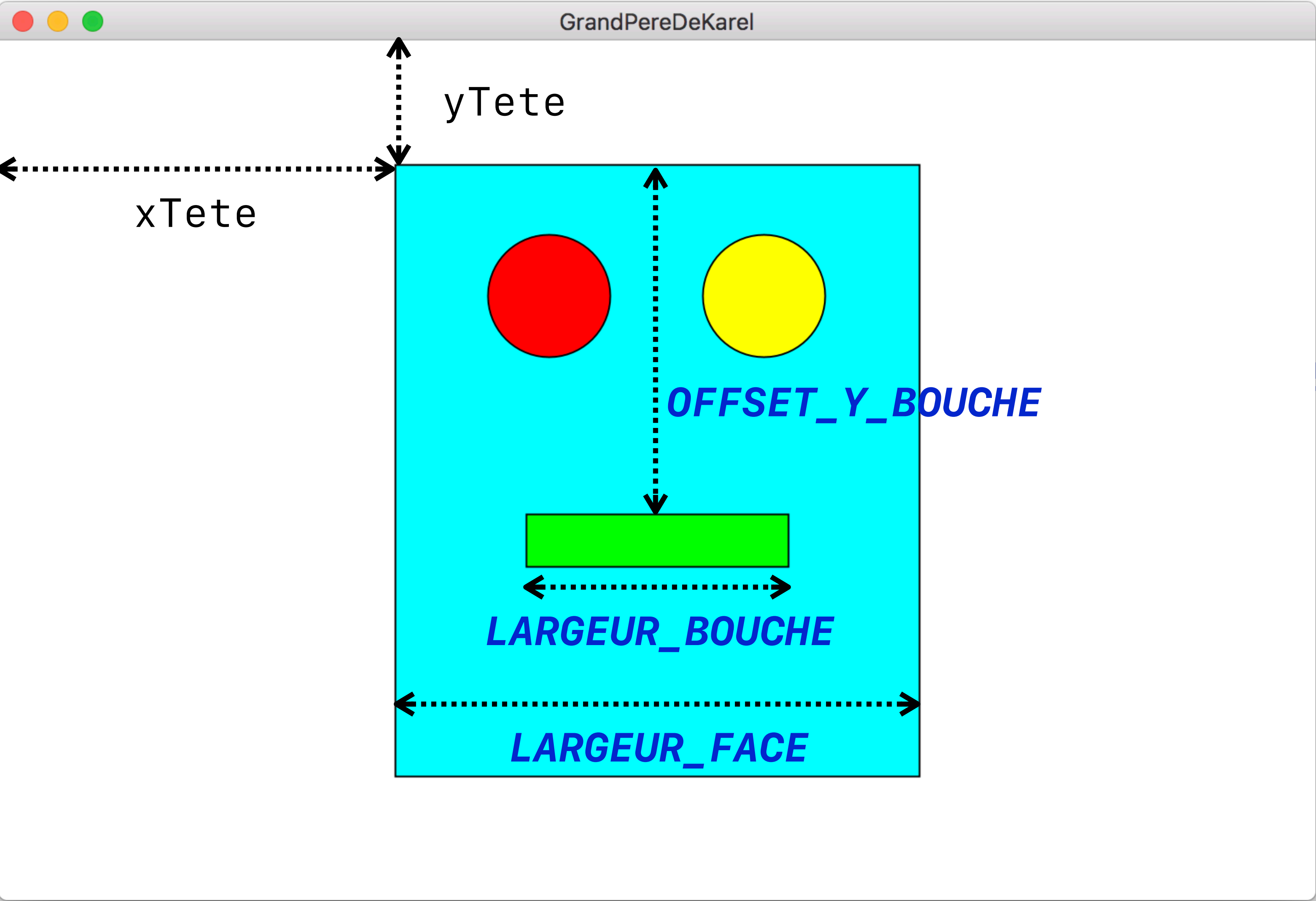
```
private void dessinerOeilGauche() {  
    double xTete = (getWidth() - LARGEUR_FACE)/2;  
    double yTete = (getHeight() - HAUTEUR_FACE)/2;  
    double distanceEntreYeux = (LARGEUR_FACE - 2*DIAMETERE_OEIL)/3;  
  
    double x = xTete + 2*distanceEntreYeux + DIAMETERE_OEIL;  
    double y = yTete + OFFSET_Y_OEIL;  
    GOval oeil = new GOval(x, y, DIAMETERE_OEIL, DIAMETERE_OEIL);  
    oeil.setFilled(true);  
    oeil.setFillColor(Color.YELLOW);  
    add(oeil);  
}
```

Grand-père de Karel

```
private void dessinerOeilDroit() {  
    double yTete = (getHeight() - HAUTEUR_FACE) / 2;  
    double xTete = (getWidth() - LARGEUR_FACE) / 2;  
  
    double espaceEntreYeux = (LARGEUR_FACE - 2 * DIAMETERE_OEIL) / 3;  
  
    double x = xTete + espaceEntreYeux;  
    double y = yTete + OFFSET_Y_OEIL;  
    GOval oeil = new GOval(x, y, DIAMETERE_OEIL, DIAMETERE_OEIL);  
    oeil.setFilled(true);  
    oeil.setFillColor(Color.RED);  
    add(oeil);  
}
```

Grand-père de Karel

```
private void  
double x  
double y  
  
double x  
double y  
GRect bo  
bouche.s  
bouche.s  
add(bouc  
  
}
```

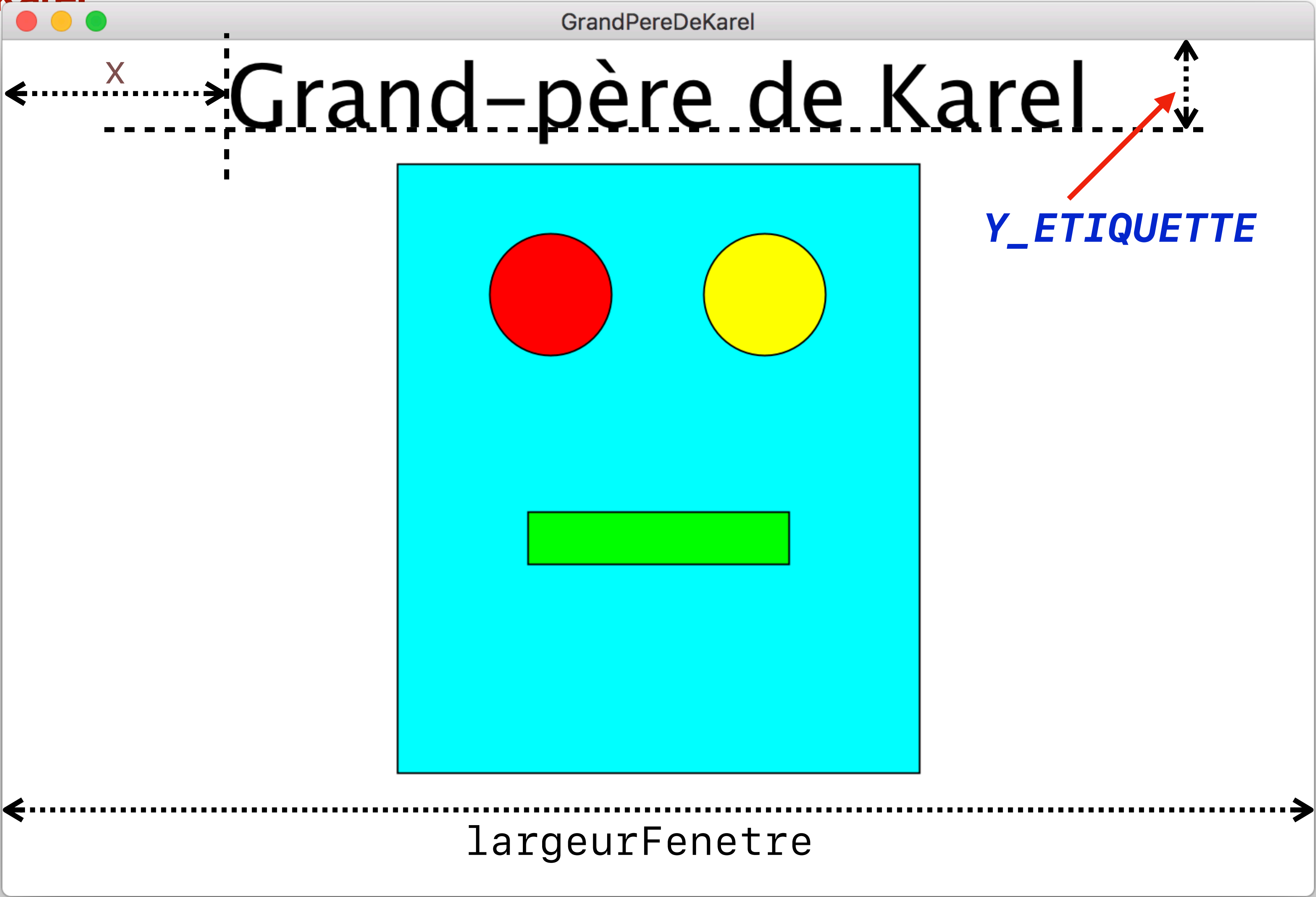


E) ;

Grand-père de Karel

```
private void dessinerBouche() {  
    double xTete = (getWidth() - LARGEUR_FACE) / 2;  
    double yTete = (getHeight() - HAUTEUR_FACE) / 2;  
  
    double x = xTete + (LARGEUR_FACE - LARGEUR_BOUCHE) / 2;  
    double y = yTete + OFFSET_Y_BOUCHE;  
    GRect bouche = new GRect(x, y, LARGEUR_BOUCHE, HAUTEUR_BOUCHE);  
    bouche.setFilled(true);  
    bouche.setFillColor(Color.GREEN);  
    add(bouche);  
}
```

Grand-père de Karel



```
private void  
GLabel  
etique  
double  
etique  
  
add(et  
  
}
```


Grand-père de Karel

```
private void dessinerEtiquette() {  
    GLabel etiquette = new GLabel("Grand-père de Karel");  
    etiquette.setFont("Courier-50");  
    double x = (getWidth() - etiquette.getWidth())/2;  
    etiquette.setLocation(x, Y_ETIQUETTE);  
  
    add(etiquette);  
}
```

Objectif de ce cours

- ☒ Variables, révision
- ☒ GraphicsProgram: Introduction et exemple
- ☒ Quelques objets graphiques communs
- ☐ Grand-père de Karel

Objectif de ce cours

- ☒ Variables, révision
- ☒ GraphicsProgram: Introduction et exemple
- ☒ Quelques objets graphiques communs
- ☒ Grand-père de Karel

Exercise: Programmer, c'est génial!

