

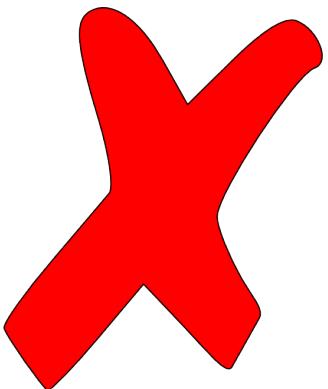


Graphiques

La programmation prend de la pratique.

Quelques problèmes normales

Réutilisation des Variables



```
16
17
18
19
20
21
22
23
```

```
public void run() {
    int counter = 0;
    for (int i = 0; i < 10; i++) {
        int counter = counter + 1;
    }
}
```

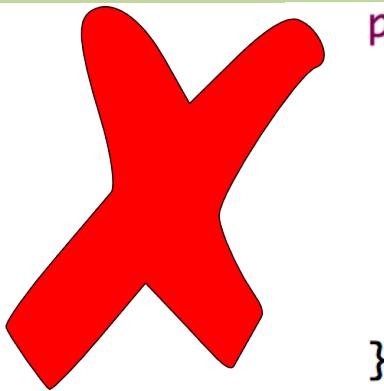
Tu dois indiquer le type d'une variable une fois.



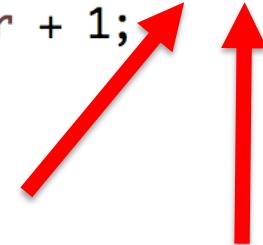
```
16
17
18
19
20
21
22
23
```

```
public void run() {
    int counter = 0;
    for (int i = 0; i < 10; i++) {
        counter = counter + 1;
    }
}
```

Où il faut mettre; les point-virgules



```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++); {  
        counter = counter + 1;  
    }  
}
```



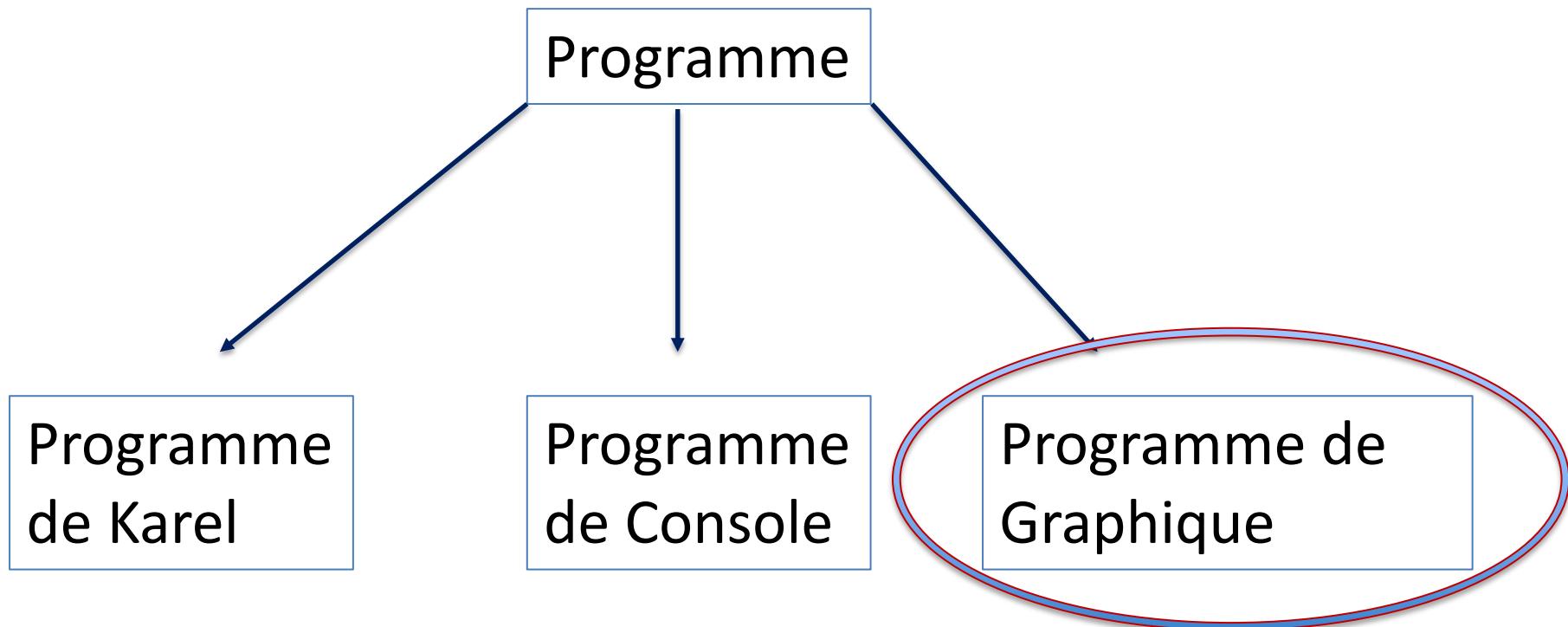
Les point-virgules ne va pas apres for,
while, if, else.

; et { ne va PAS ensemble

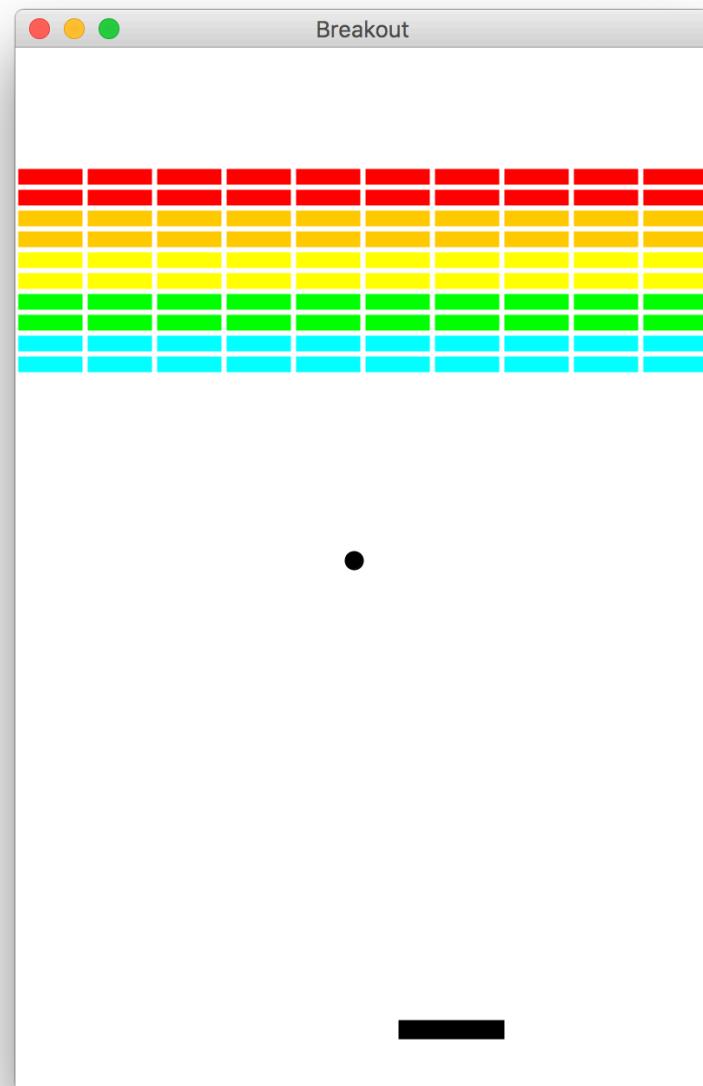
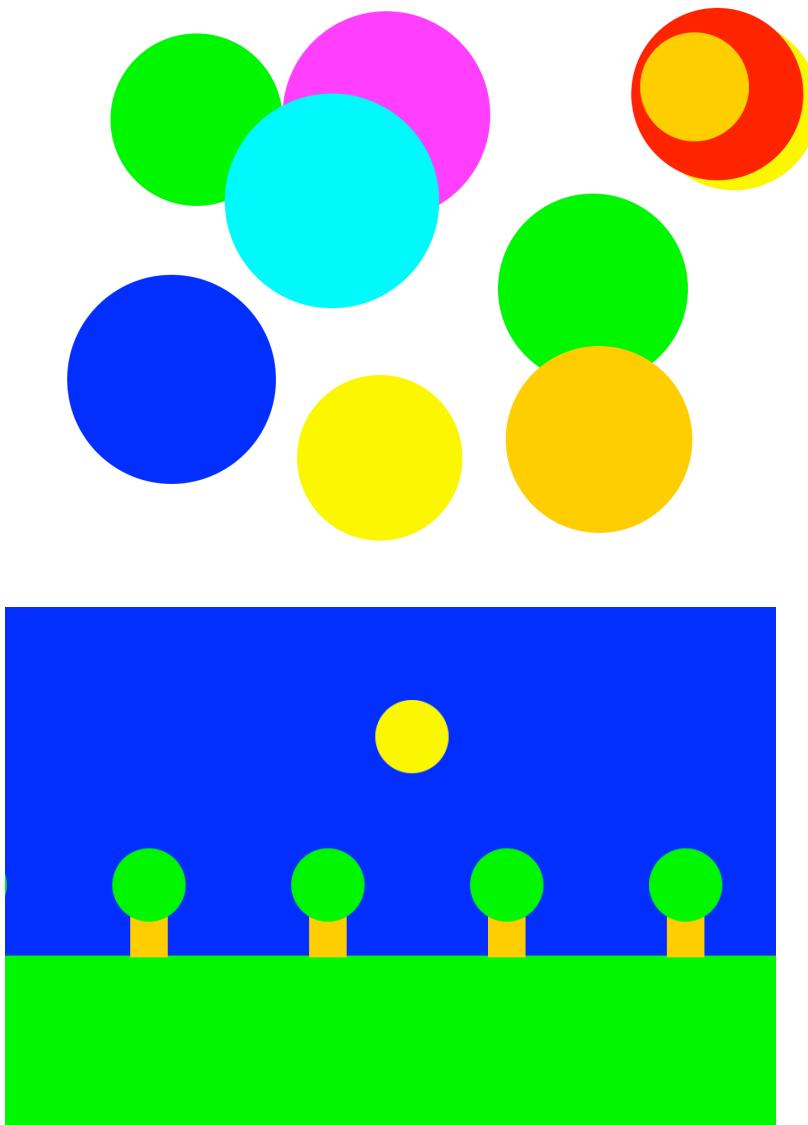


```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++) {  
        counter = counter + 1;  
    }  
}
```

Après les Programmes de Console



Après les Programmes de Console

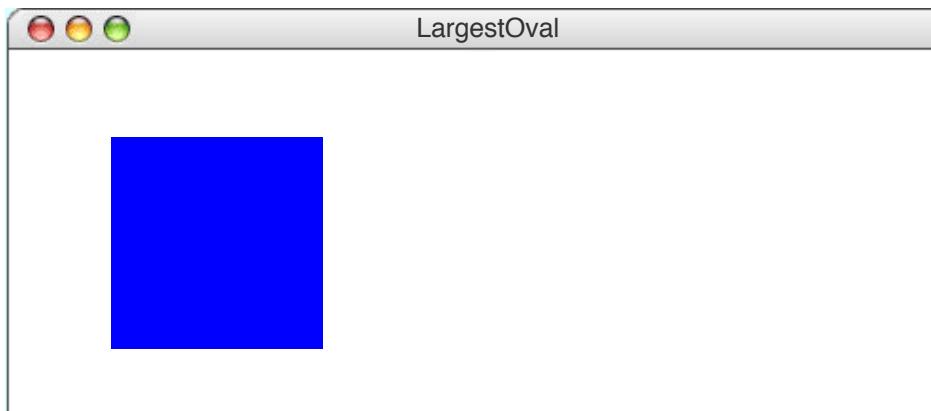


Notre Première Programme de Graphique

GRect est un variable qui contien un rectangle.

Par exemple, le suivant `run` methode montre un rectangle

```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```



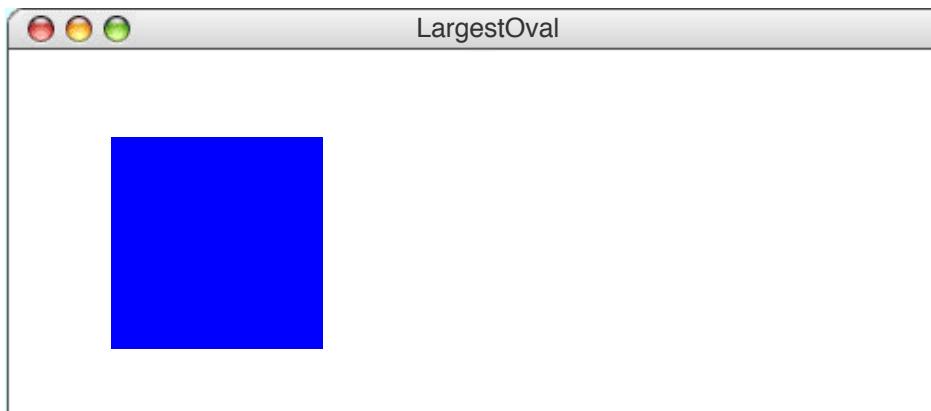
Rect =

Notre Première Programme de Graphique

GRect est un variable qui contien un rectangle.

Par exemple, le suivant `run` methode montre un rectangle

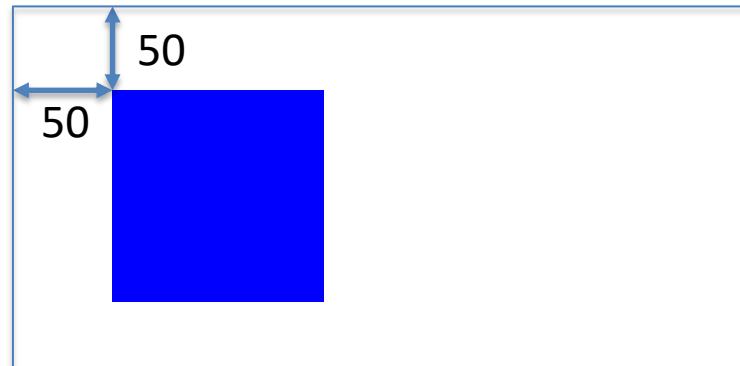
```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```



Rect =

Notre Première Programme de Graphique

```
public void run() {  
    // Créer un 200x250 GRect à (50, 50)  
    GRect rect = new GRect(50, 50, 200, 200);  
  
    // Définir des propriétés  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
  
    // Ajouter à l'écran  
    add(rect);  
}
```



Notre Première Programme de Graphique

```
public void run() {  
    // Créer un 200x250 GRect à (50, 50)  
    GRect rect = new GRect(50, 50, 200, 200);  
  
    // Définir des propriétés  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
  
    // Ajouter a l'écran  
    add(rect);  
}
```

Notre Première Programme de Graphique

```
public void run() {  
    // Créer un 200x250 GRect à (50, 50)  
    GRect rect = new GRect(50, 50, 200, 200);  
  
    // Définir des propriétés  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
  
    // Ajouter à l'écran  
    add(rect);  
}
```

Notre Première Programme de Graphique

```
public void run() {  
    // Créer un 200x250 GRect à (50, 50)  
    GRect rect = new GRect(50, 50, 200, 200);  
  
    // Définir des propriétés  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
  
    // Ajouter a l'écran  
    add(rect);  
}  
}
```

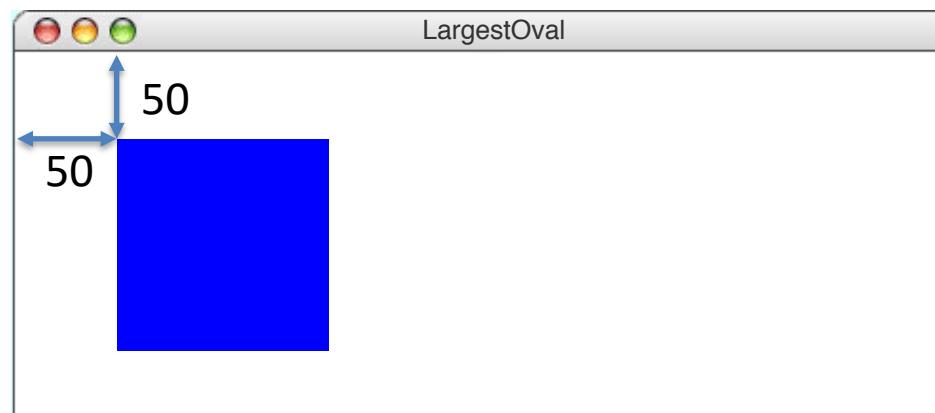
GRect

GRect est un variable qui contien un rectangle.

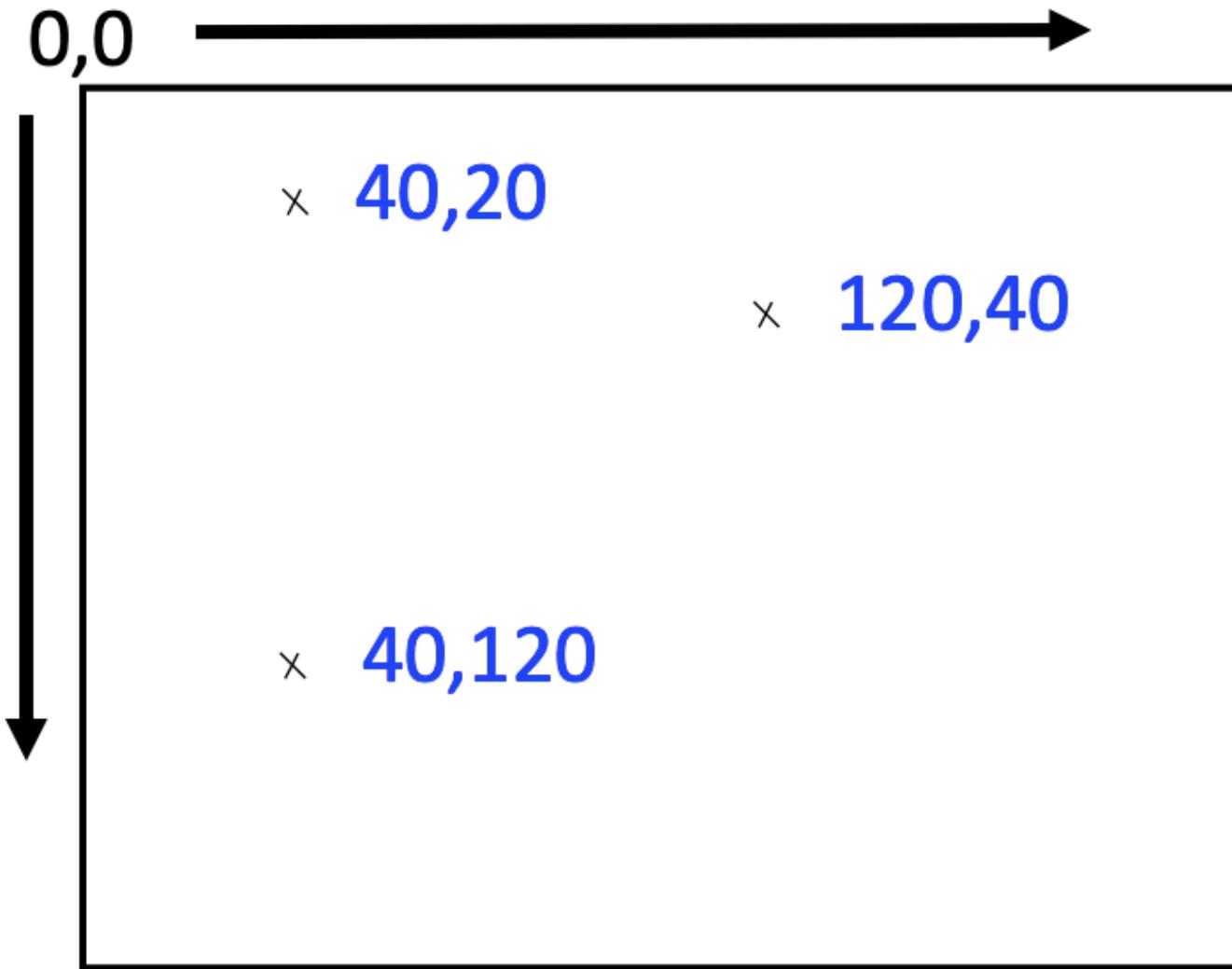
Par exemple, le suivant `run` methode montre un rectangle

```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```

Les coordonnées
d'un rectangle sont
le coin supérieur
gauche.



Les Coordonnées Graphiques



Les Objets Graphiques

GRect

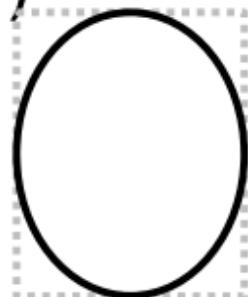
(x, y)



(x+w,
y+h)

GOval

(x, y)



(x+w,
y+h)

GLine

(x₁, y₁)

(x₂, y₂)

GLabel

Hello there!

GImage



GArc



GRoundRect



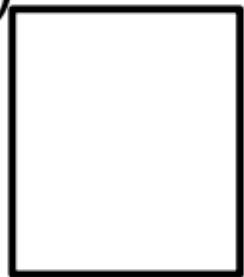
GPolygon



Les Objets Graphiques

GRect

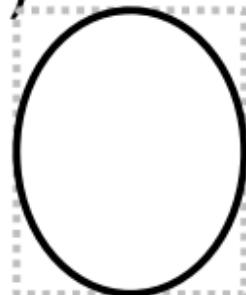
(x, y)



(x+w,
y+h)

GOval

(x, y)



(x+w,
y+h)

GLine

(x₁, y₁)

(x₂, y₂)

GLabel

Hello there!

GImage



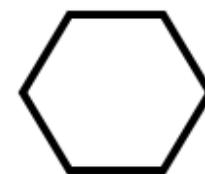
GArc



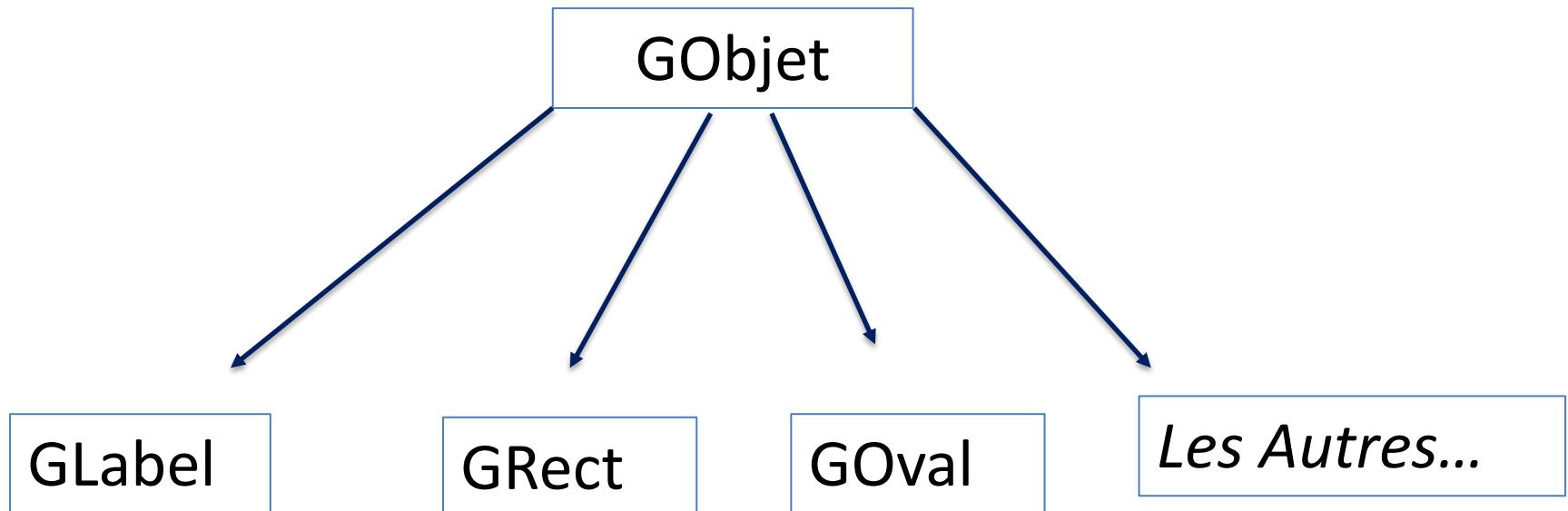
GRoundRect



GPolygon



Après les Programmes de Console



```
GRect monRect = new GRect(50, 50, 350, 270);
```

Primitives vs. Objects

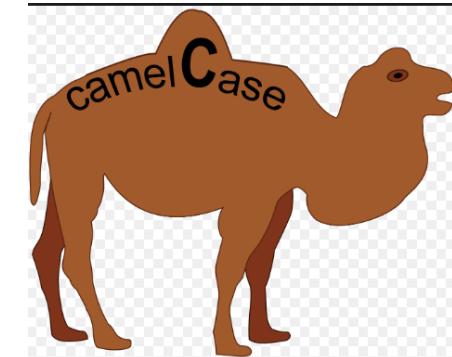
Types de Variables Primitives

int
double
char
boolean

Types de Variables Objets

GRect
GOval
GLine
GLabel

Chameau



Variables des objets:

1. Avoir des types majuscule chameaux
2. Vous pouvez appeler des méthodes sur eux
3. Sont construits en utilisant **new**

Méthodes pour des Objets

Nous manipulons des objets graphiques en appelant méthodes sur eux:

objet.méthode(paramètres);

Receveur

Message

Méthodes pour des Objets

Nous manipulons des objets graphiques en appelant méthodes sur eux:

objet . méthode (paramètres) ;

Qui?

Quoi?

Quoi exactement?

Exemple:

rect . setColor (Color . RED) ;

Méthodes pour GObjet

Les opérations suivantes peuvent être utilisées pour toutes les GObjets

object.setColor(color)

Définit la couleur de l'objet sur la couleur spécifiée.

object.setLocation(x, y)

Modifie la position de l'objet en point (x, y)

object.move(dx, dy)

Déplace l'objet à l'écran en ajoutant dx et dy à sa position actuelle

object.getWidth()

Retourne la largeur de l'objet.

object.getHeight()

Retourne la hauteur de l'objet.

Couleurs

- Specifié par constantes de Color [Couleur en français]

Ecrivez Color.NOM ou NOM est une de:

BLACK	BLUE	CYAN	DARK_GRAY	GRAY
GREEN	LIGHT_GRAY	MAGENTA	ORANGE	PINK
RED	WHITE	YELLOW		

```
rect.setColor(Color.GREEN);
```

Ou bien, utiliser Rouge-Vert-Bleu
(Red, Green, Blue RGB) VALUES OF 0–255
new Color(*red*, *green*, *blue*)

-Exemple:

```
rect.setColor(new Color(192, 128, 64));
```

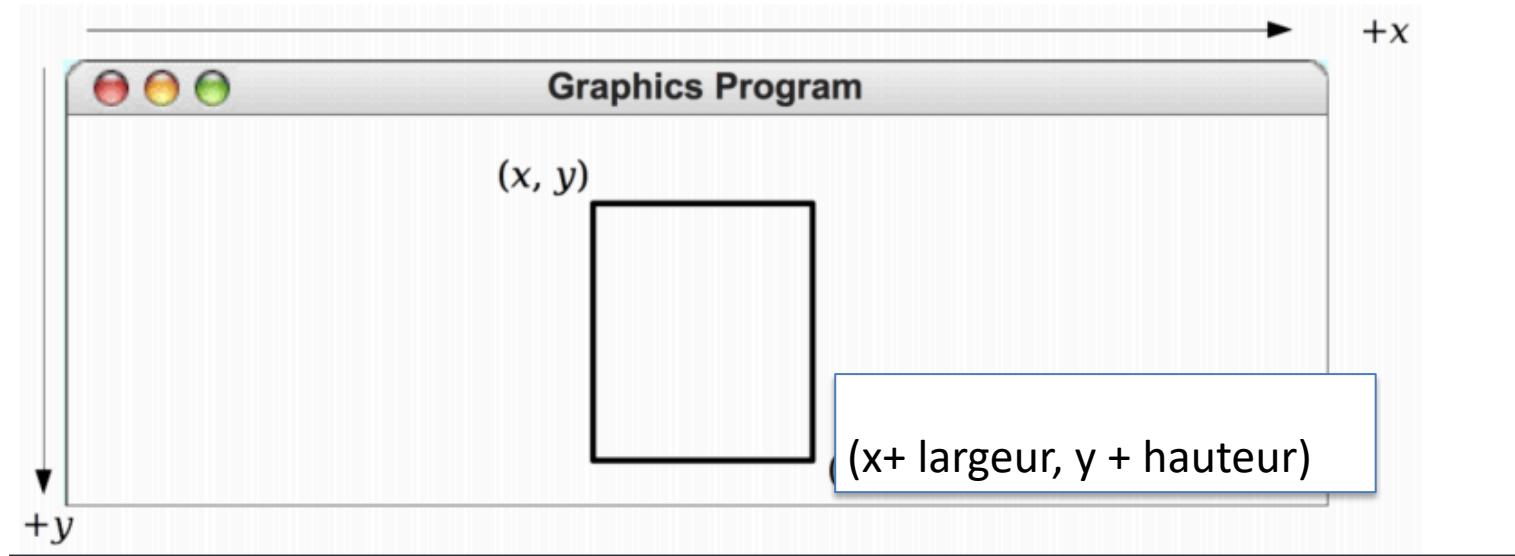
GRect

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

-Crée un rectangle avec la largeur et la hauteur données, dont le coin supérieur gauche est à (x, y)

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

-Comme ci-dessus, mais la valeur par défaut est $(x, y)=(0, 0)$



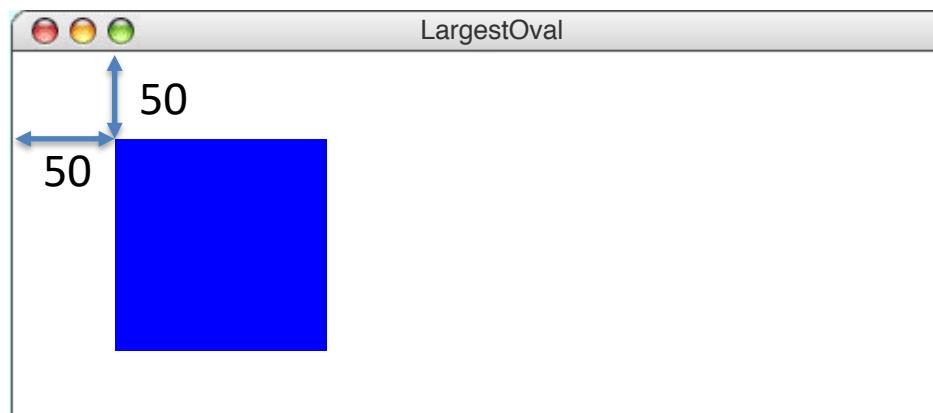
GRect

GRect est un variable qui contien un rectangle.

Par exemple, le suivant `run` methode montre un rectangle

```
public void run() {  
    GRect rect = new GRect(200, 200);  
    rect.setFilled(true);  
    rect.setColor(Color.BLUE);  
    add(rect, 50, 50);  
}
```

Les coordonnées
d'un rectangle sont
le coin supérieur
gauche.



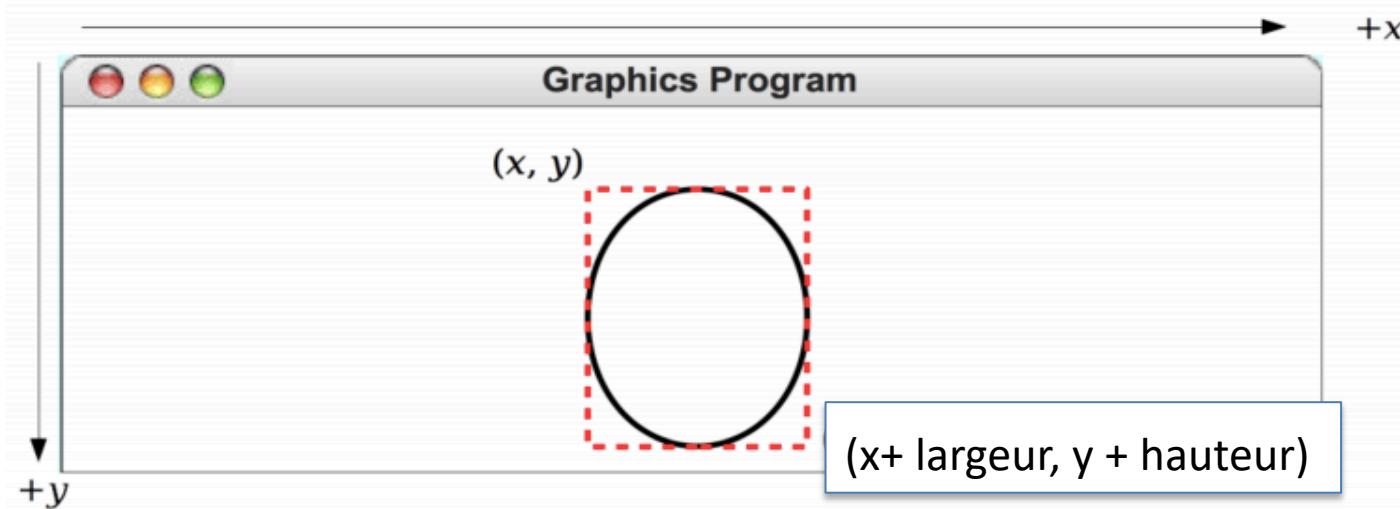
GOval

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

-Crée un ovale avec la largeur et la hauteur données, dont le coin supérieur gauche est à (x, y)

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

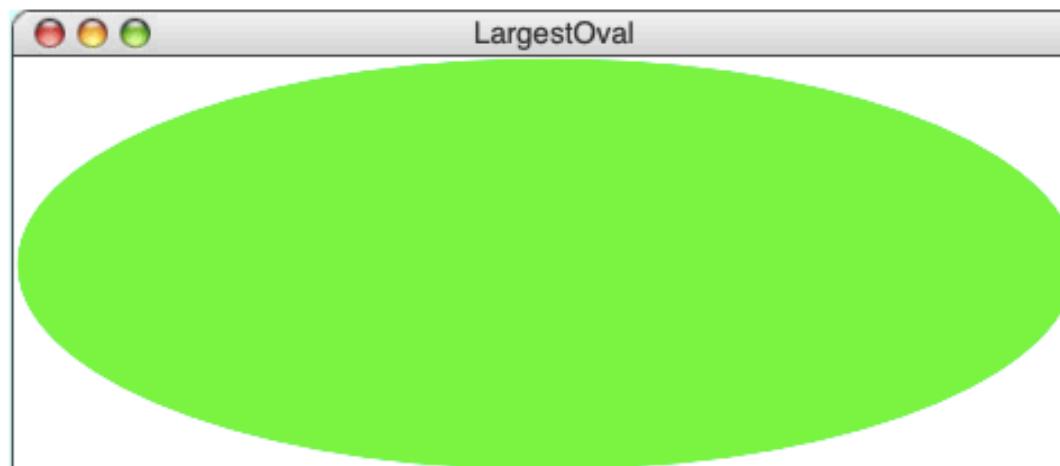
-Comme ci-dessus, mais la valeur par défaut est $(x, y)=(0, 0)$



GOval

Par exemple, le suivant `run` créer le plus large ovale qui s'inscrit dans l'écran

```
public void run() {  
    GOval oval = new GOval(getWidth(), getHeight());  
    oval.setFilled(true);  
    oval.setColor(Color.GREEN);  
    add(oval, 0, 0);  
}
```



GRect et GOval

Les méthodes partagées par GRect et Goval classes:

object.setFilled(fill)

Si fill est vrai, remplit l'intérieur de l'objet;
si faux, ne montre que le contour.

object.setFillColor(color)

Définit la couleur utilisée pour remplir l'intérieur,
qui peut être différente de la frontière.

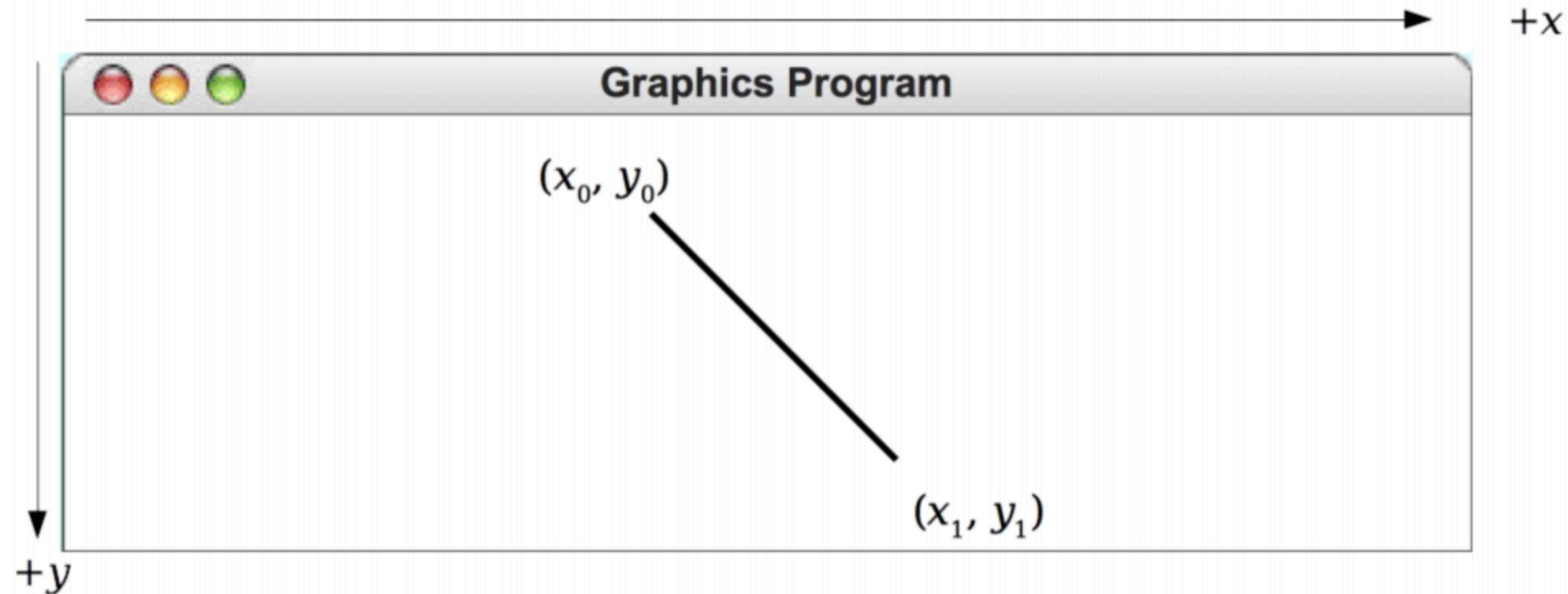
object.setSize(largeur, hauteur)

Définit la taille de l'objet à la largeur et à la hauteur données

GLine

```
new GLine(x0, y0, x1, y1)
```

Crée une ligne allant de (x_0, y_0) à (x_1, y_1)



GLabel

new GLabel("votre text ici", x, y)

-Crée une étiquette avec le texte donné, dont la ligne de base commence à (x, y). PAS positionné selon le coin en haut à gauche!

new GLabel("votre text ici")

-Comme ci-dessus, mais la valeur par défaut est (x, y) = (0, 0)



Méthodes pour GLabel

Les méthodes spécifique pour le classe de **Glabel** class.

`label.setFont(font)`

Définit la font utilisée pour montrer le label

Le font est normalement specifier par un string dans la forme:

`"famille-style-taille"`

famille – est le nom d'un familles de font

style – est PLAIN, **BOLD**, *ITALIC*, ou **BOLDITALIC**

size - est un nombre entier qui dit la taille des lettres

GLabel

Un variable qui représente le texte.

```
Public class HelloProgram extends GraphicsProgram {  
    public void run() {  
        GLabel label = new GLabel("hello, world", 100, 75);  
        label.setFont("SansSerif-36");  
        label.setColor(Color.RED);  
        add(label);  
    }  
}
```



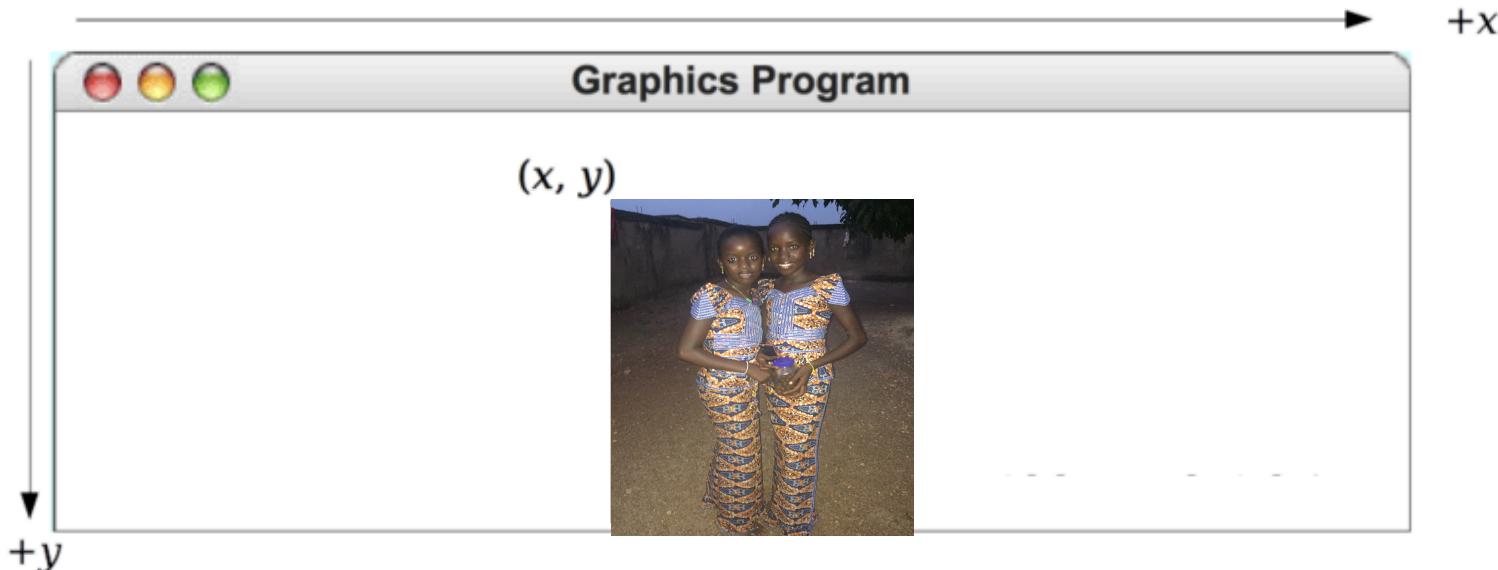
GImage

new GImage("votre docier ici", x, y)

-Crée une image de image/docier dont le coin supérieur gauche est à (x, y)

new GImage("votre docier ici")

-Comme ci-dessus, mais la valeur par défaut est (x, y) = (0, 0)



Méthodes pour GImage

Les méthodes spécifique pour le classe de **Glabel** class.

objet.setSize(largeur, hauteur)

Définit la taille de l'objet à la largeur et à la hauteur données

Le font est normalement specifier par un string dans la forme:

"famille-style-taille"

famille – est le nom d'un familles de font

style – est PLAIN, **BOLD**, *ITALIC*, ou **BOLDITALIC**

size - est un nombre entier qui dit la taille des lettres

Graphics Methods

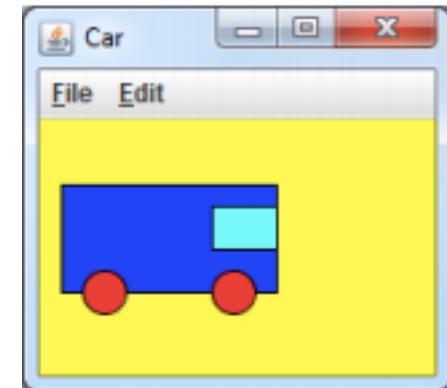
<code>add (<i>object</i>)</code>	Adds the object to the canvas at the front of the stack
<code>add (<i>object</i>, <i>x</i>, <i>y</i>)</code>	Moves the object to (<i>x</i> , <i>y</i>) and then adds it to the canvas
<code>remove (<i>object</i>)</code>	Removes the object from the canvas
<code>removeAll ()</code>	Removes all objects from the canvas
<code>getElementAt (<i>x</i>, <i>y</i>)</code>	Returns the frontmost object at (<i>x</i> , <i>y</i>), or <code>null</code> if none
<code>getWidth ()</code>	Returns the width in pixels of the entire canvas
<code>getHeight ()</code>	Returns the height in pixels of the entire canvas
<code>setBackground (<i>c</i>)</code>	Sets the background color of the canvas to <i>c</i> .
<code>pause (<i>milliseconds</i>)</code>	Pauses the program for the specified time in milliseconds
<code>waitForClick ()</code>	Suspends the program until the user clicks the mouse

Graphics Methods

<code>add (<i>object</i>)</code>	Adds the object to the canvas at the front of the stack
<code>add (<i>object</i>, <i>x</i>, <i>y</i>)</code>	Moves the object to (<i>x</i> , <i>y</i>) and then adds it to the canvas
<code>remove (<i>object</i>)</code>	Removes the object from the canvas
<code>removeAll ()</code>	Removes all objects from the canvas
<code>getElementAt (<i>x</i>, <i>y</i>)</code>	Returns the frontmost object at (<i>x</i> , <i>y</i>), or <code>null</code> if none
<code>getWidth ()</code>	Returns the width in pixels of the entire canvas
<code>getHeight ()</code>	Returns the height in pixels of the entire canvas
<code>setBackground (<i>c</i>)</code>	Sets the background color of the canvas to <i>c</i> .
<code>pause (<i>milliseconds</i>)</code>	Pauses the program for the specified time in milliseconds
<code>waitForClick ()</code>	Suspends the program until the user clicks the mouse

Pratique: La Voiture

Ecrivez un programme graphique nommé Voiture qui dessine une figure qui ressemble à une voiture.



- Pneus rouges à (20, 70) et (80, 70), taille 20x20
- Pare-brise cyan à (80, 40), taille 30x20
- Corps bleu à (10, 30), taille 100x50
- Fond jaune

Solution: La Voiture

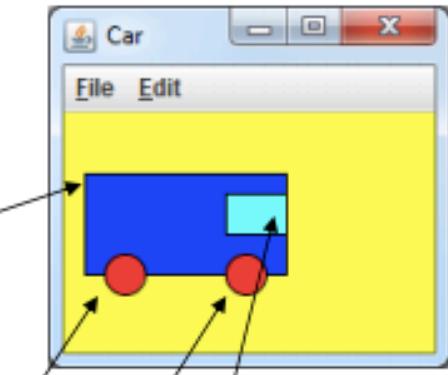
```
// When 2 shapes occupy the same pixels, the last one drawn "wins"
public class Car extends GraphicsProgram {
    public void run() {
        setBackground(Color.YELLOW);

        GRect body = new GRect(10, 30, 100, 50);
        body.setFilled(true);
        body.setFillColor(Color.BLUE);
        add(body);

        GOval wheel1 = new GOval(20, 70, 20, 20);
        wheel1.setFilled(true);
        wheel1.setFillColor(Color.RED);
        add(wheel1);

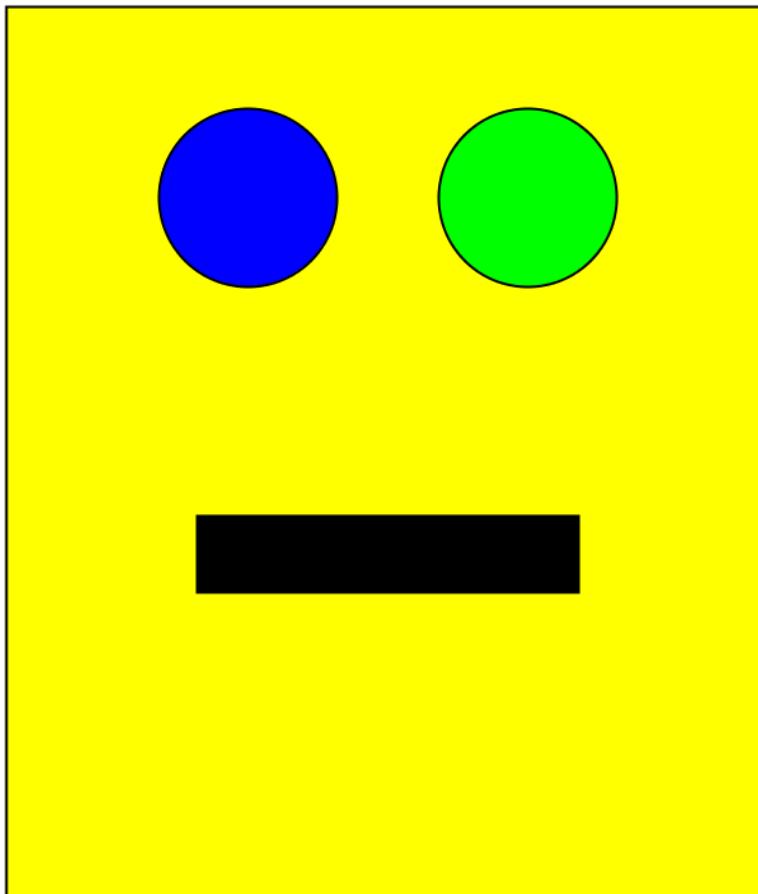
        GOval wheel2 = new GOval(80, 70, 20, 20);
        wheel2.setFilled(true);
        wheel2.setFillColor(Color.RED);
        add(wheel2);

        GRect windshield = new GRect(80, 40, 30, 20);
        windshield.setFilled(true);
        windshield.setFillColor(Color.CYAN);
        add(windshield);
    }
}
```

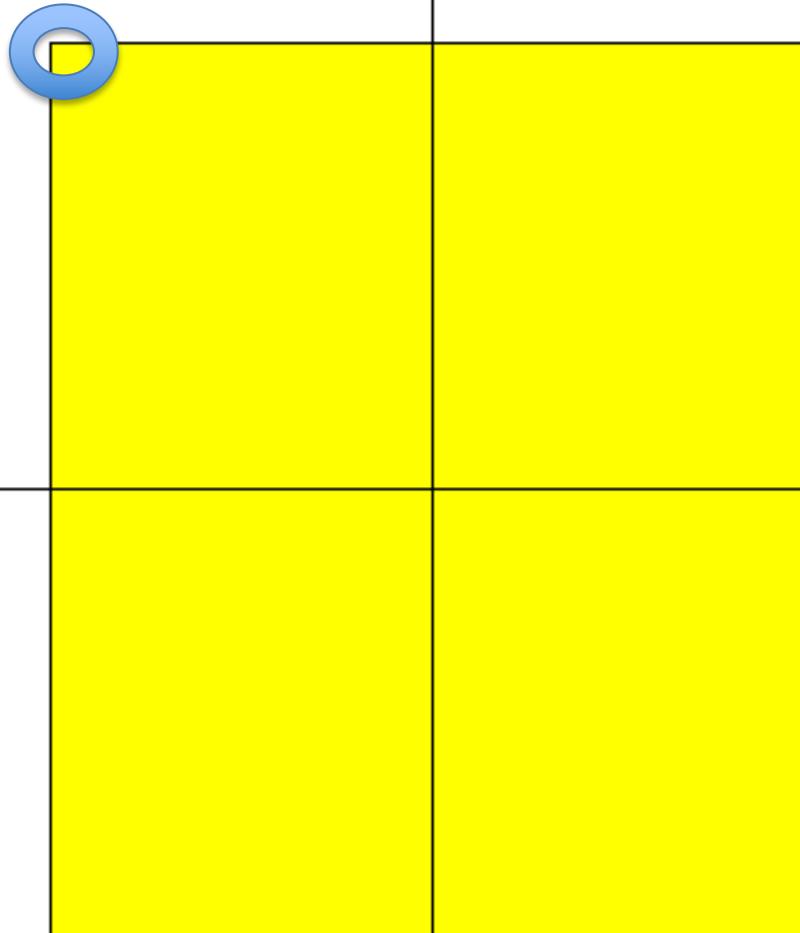


Le Grandpère de Karel

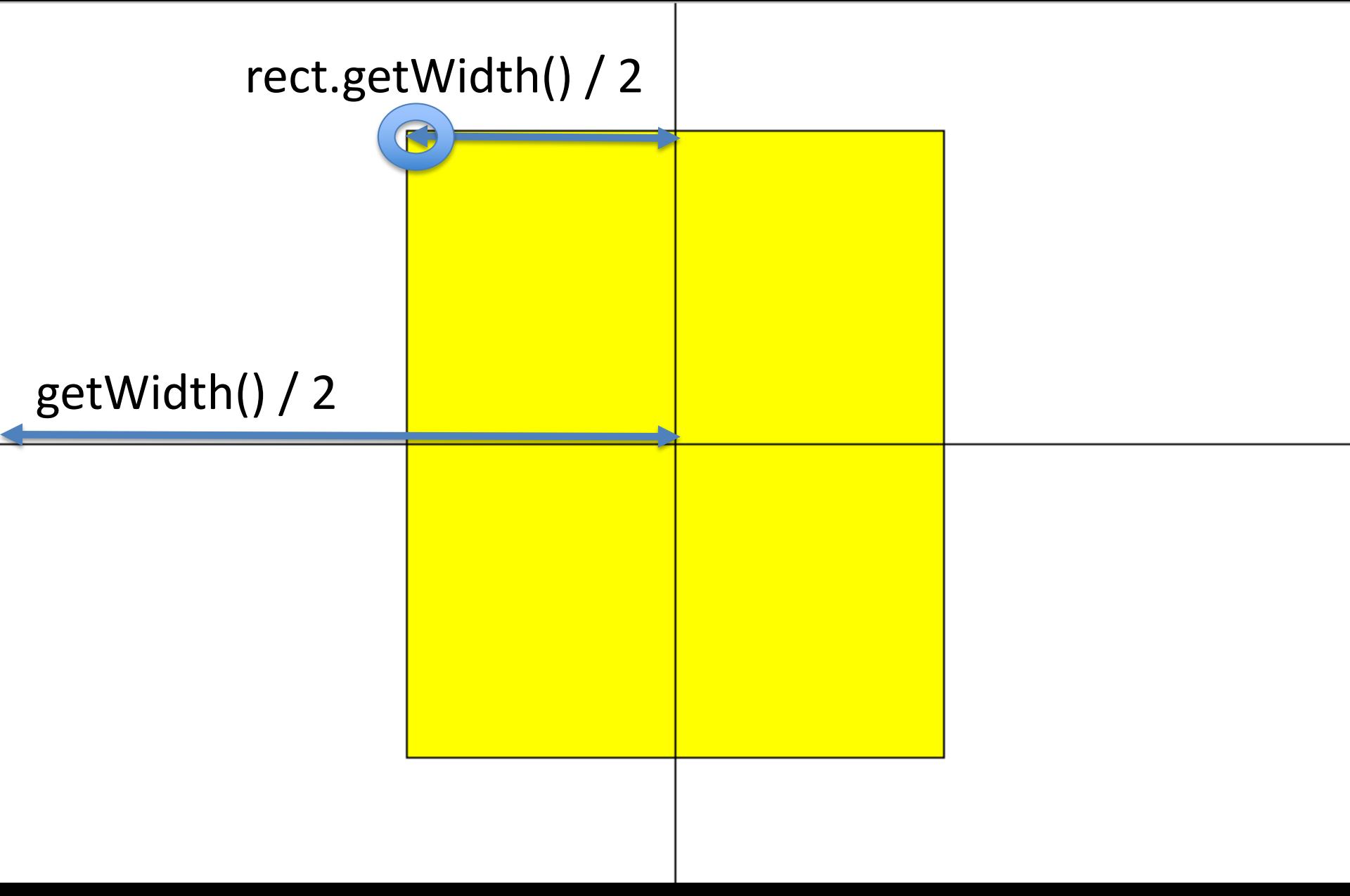
Robot Face



Karel's Grandpa

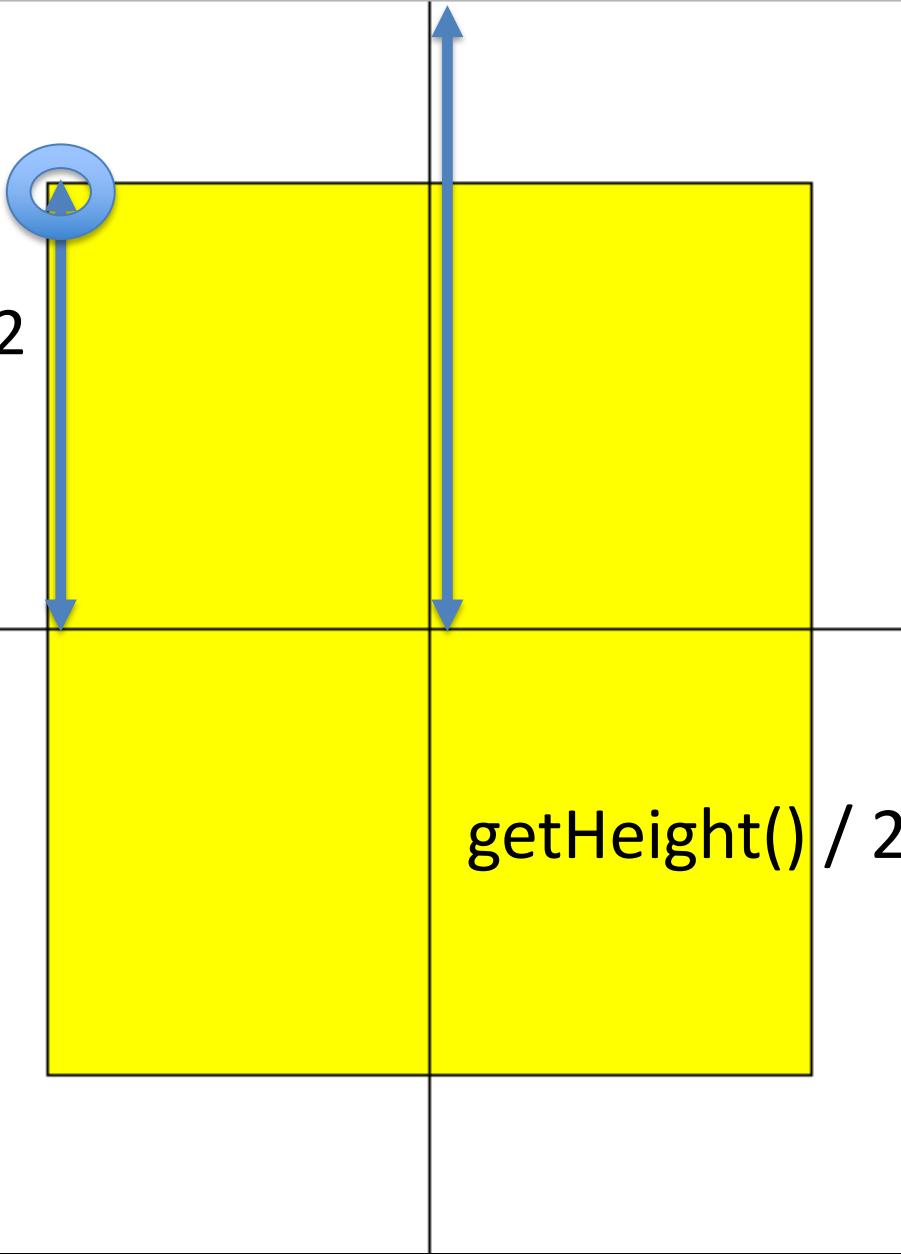


Karel's Grandpa



Karel's Grandpa

`rect.getHeight() / 2`

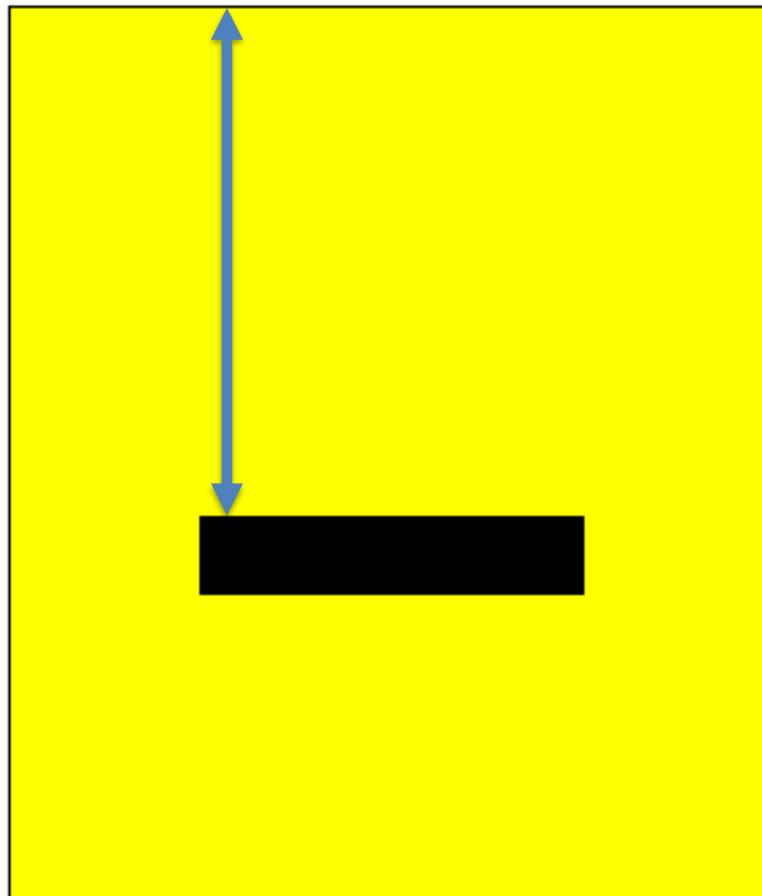


Constants

```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```

Karel's Grandpa

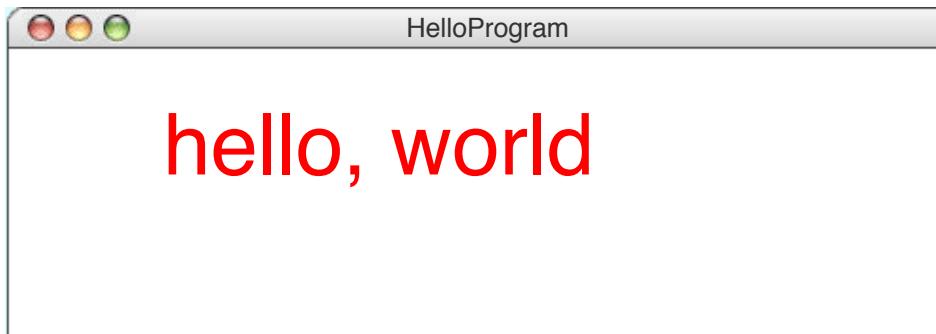
```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```



GLabel

Un variable qui represent le text

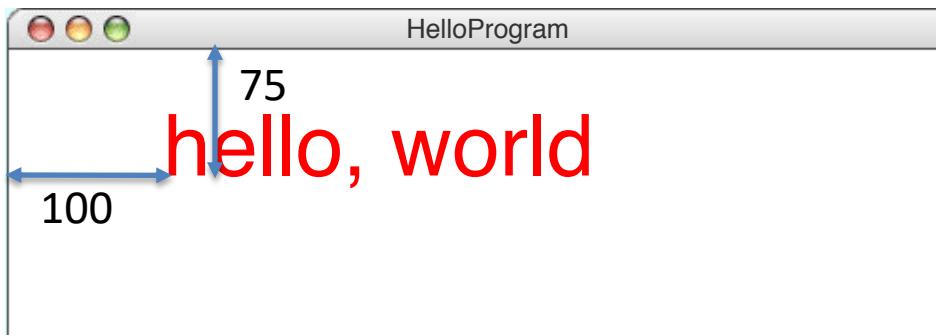
```
public class HelloProgram extends GraphicsProgram {  
    public void run() {  
        GLabel label = new GLabel("hello, world", 100, 75);  
        label.setFont("SansSerif-36");  
        label.setColor(Color.RED);  
        add(label);  
    }  
}
```



GLabel

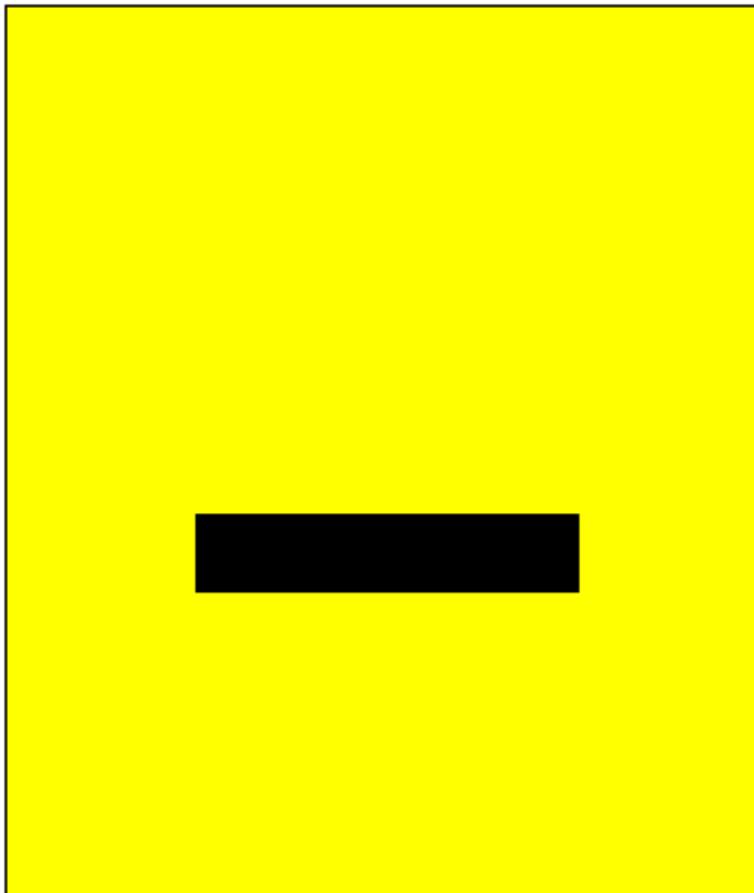
Un variable qui represent le text

```
public class HelloProgram extends GraphicsProgram {  
    public void run() {  
        GLabel label = new GLabel("hello, world", 100, 75);  
        label.setFont("SansSerif-36");  
        label.setColor(Color.RED);  
        add(label);  
    }  
}
```



Grandpere de Karel

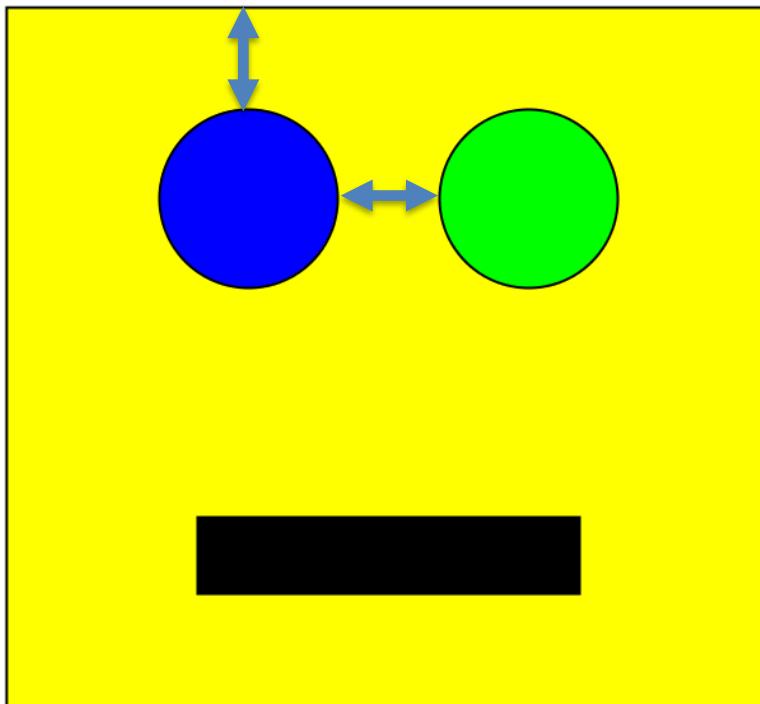
Robot Face



```
/*The distance from the top of the screen to the base of the label:*/  
private static final int LABEL_Y = 50;
```

Grandpère de Karel

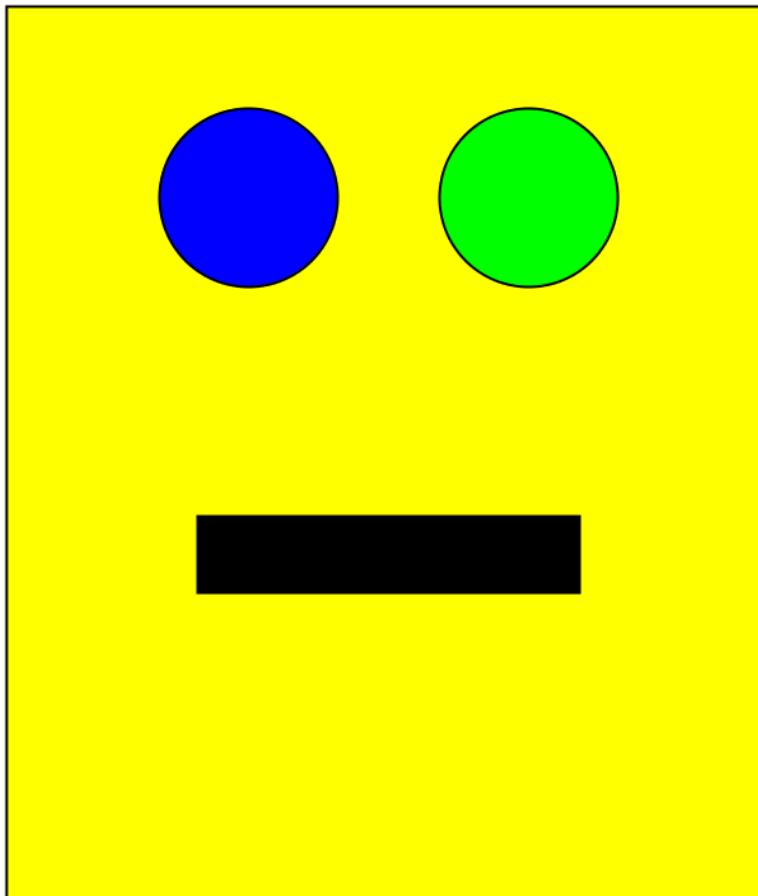
Robot Face



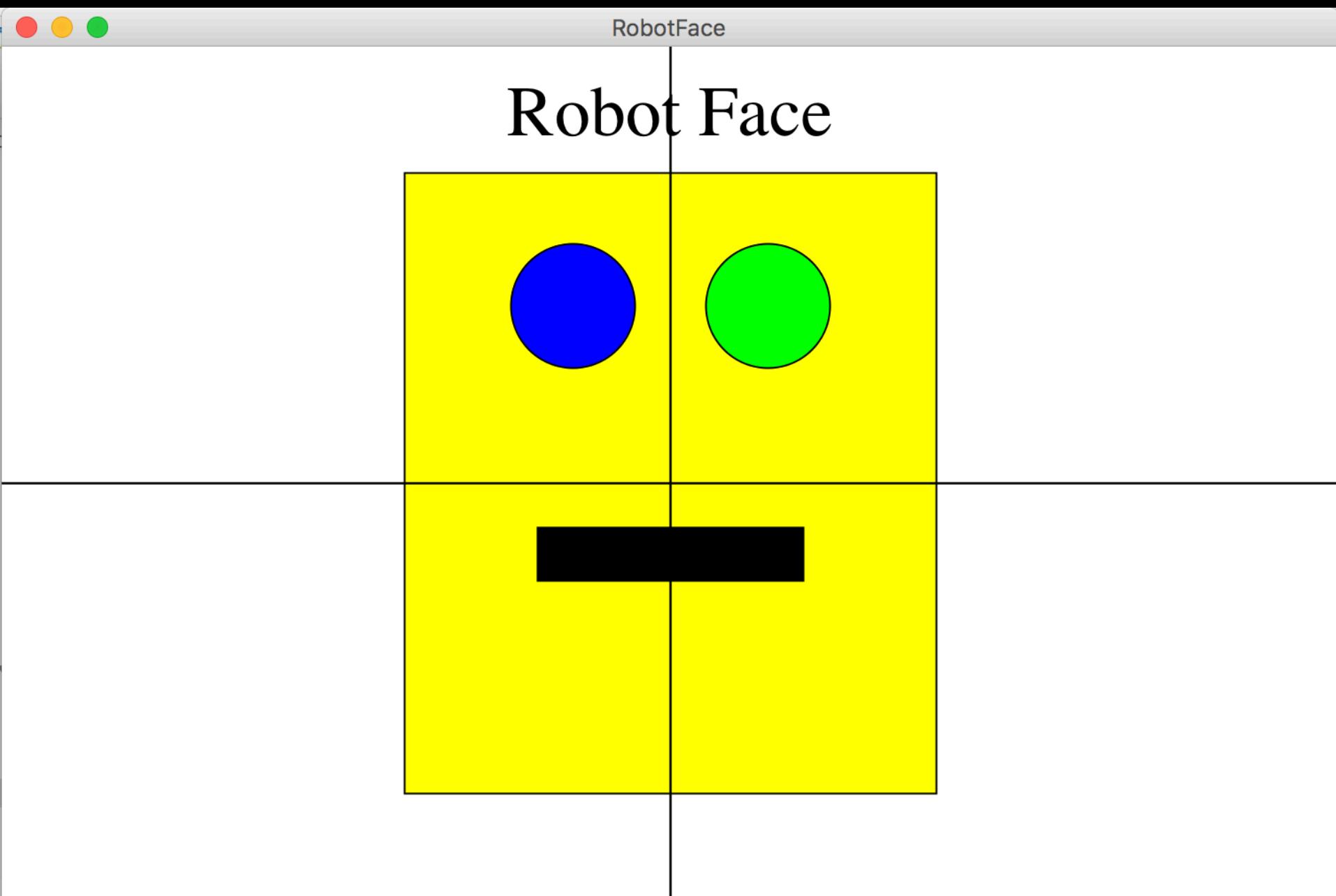
```
/*The distance from the top of the head to the top of the eyes:*/  
private static final int EYE_Y_OFFSET = 40;  
  
/*The distance in between the two eyes:*/  
private static final int EYE_X_SEPARATION = 40;
```

Grandpère de Karel

Robot Face



Grandpère de Karel



Reference

Constructors

new GLabel(String text) or new GLabel(String text, double x, double y)
Creates a new GLabel object; the second form sets its location as well.
new GRect(double x, double y, double width, double height)
Creates a new GRect object; the x and y parameters can be omitted and default to 0.
new GOval(double x, double y, double width, double height)
Creates a new GOval object; the x and y parameters can be omitted and default to 0.
new GLine(double x1, double y1, double x2, double y2)
Creates a new GLine object connecting (x1, y1) and (x2, y2).

Methods common to all graphical objects

void setLocation(double x, double y)
Sets the location of this object to the specified coordinates.
void move(double dx, double dy)
Moves the object using the displacements dx and dy .
double getWidth()
Returns the width of the object.
double getHeight()
Returns the height of the object.
void setColor(Color c)
Sets the color of the object.

Methods available for GRect and GOval only

void setFilled(boolean fill)
Sets whether this object is filled (true means filled, false means outlined).
boolean isFilled()
Returns true if the object is filled.
void setFillColor(Color c)
Sets the color used to fill this object. If the color is null , filling uses the color of the object.

Methods available for GLabel only

void setFont(String fontName)
Sets the font, as described in Chapter 5.
double getAscent()
Returns the height above the text baseline.
double getDescent()
Returns the height below the text baseline.