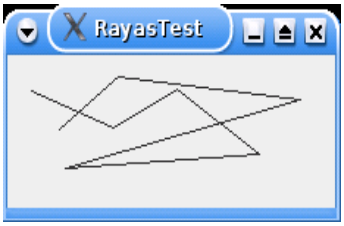
## Certamen Final

1.

Se tiene una aplicación Java que permite dibujar rayas rectas con el mouse. Cada vez que se presiona el mouse se agrega a lo ya hecho una línea entre el último punto y el actual. El primer punto sólo define el extremo de la línea. El código que implementa tal aplicación es:




```
public class RayasTest {
   public static void main(String[] args) {
      RayasFrame frame = new RayasFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}

class RayasFrame extends Jframe {
   public RayasFrame()   {
      setTitle("RayasTest");
      setSize(250, 150);

      RayasPanel panel = new RayasPanel();
      Container contentPane = getContentPane();
      contentPane.add(panel);
   }
}

class RayasPanel extends Jpanel {
   public RayasPanel()  {
      puntos = new ArrayList();
      addMouseListener(new MouseHandler());
   }

    public void paintComponent(Graphics g)   {
       super.paintComponent(g);
       Graphics2D g2 = (Graphics2D)g;

       for (int i = 0; i < puntos.size()-1; i++)
           g2.draw(new Line2D.Double((Point)puntos.get(i),
                                     (Point)puntos.get(i+1)));
    }

    private ArrayList puntos;

    private class MouseHandler extends MouseAdapter    {
       public void mouseClicked(MouseEvent event)     {
           puntos.add(event.getPoint());
           repaint();
       }
    }
}
```

a) Se pide que usted desarrolle un applet que haga lo mismo que hace esta aplicación y que permita a los usuarios de su applet, definir en el archivo html el color de las líneas. Ayuda: puede hacer uso del método setColor de la clase Graphics, ejemplo: g.setColor(Color.red);

**17 pts.**

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.awt.geom.*;
import javax.swing.*;

public class RayasApplet extends Japplet   4/4
{
```

```
   public void init()  4/4
   {
      String color=getParameter("Color");  3/3
      RayasPanel panel = new RayasPanel(color);
      Container contentPane = getContentPane();
      contentPane.add(panel);
   }
}

class RayasPanel extends JPanel
{
   public RayasPanel(String color)  3/3 paso de info a clase
   {
       this.color=Color.decode(color); // cualquier cosa aquí es aceptable.
      puntos = new ArrayList();
      addMouseListener(new MouseHandler());
   }

    public void paintComponent(Graphics g)
    {
      super.paintComponent(g);
      Graphics2D g2 = (Graphics2D)g;
      g.setColor(color);                3/3
      for (int i = 0; i < puntos.size()-1; i++)
          g2.draw(new Line2D.Double((Point)puntos.get(i),
                                    (Point)puntos.get(i+1)));
    }

    private ArrayList puntos;
    private Color color;
    private class MouseHandler extends MouseAdapter
    {
      public void mouseClicked(MouseEvent event)
      {
          puntos.add(event.getPoint());
          repaint();
      }
    }
}
```

b) Muestre el contenido de un archivo html que use su applet con líneas rojas. 8 pts.

```
<html>
 <head>
  <title>
   "Rayas Applet"
  </title>
</head>
<body>
<applet code="RayasApplet.class" width="300" height="400">
<param name="Color" value="0xff0000"/>
</applet>
</body>
</html>
```

2. Implemente en C++ la clase Moneda cuyo prototipo es el siguiente:

```
#include <iostream>
using namespace std;
typedef unsigned int uint;  // enteros sin signo
class Moneda {
public:
 Moneda();  // valor inicial cero
 Moneda(uint diezMil, uint cincoMil, uint dosMil, uint unMil);
       // cada argumento indica el número de billetes.
 // ... otros métodos

private:
  // ????
};
```

El modelo es simple y sólo considera los valores discretos mostrados en el segundo constructor.

Complete el prototipo e implemente el prototipo para que sea posible hace cosas como:
Modena x, y(3, 2, 2, 2), z(2, 1, 3, 4);
x = y+z;
x=2*z;
x= 2*y-z;
cout << x;  // arroja el mínimo número de billetes que representa el valor,
            //  cantidades negativas ponerlas entre paréntesis.

**En archivo Moneda.h tendríamos:**

```
#ifndef MONEDA_H
#define MONEDA_H

#include <iostream>

typedef unsigned int uint;  // enteros sin signo

class Moneda {
 public:
  Moneda();  // valor inicial cero
  Moneda(uint diezMil, uint cincoMil, uint dosMil, uint unMil);
  // cada argumento indica el nmero de billetes.
  Moneda operator+(Moneda m);         2/2
  Moneda operator-(Moneda m);         2/2

  friend ostream& operator<< (ostream &os, const Moneda &m);//permite cout<<"Moneda"
3/3
  friend Moneda operator* (int i, const Moneda &m); // permite 2*Moneda"   1/1

 private:
  uint miles;   2/2
};

#endif
```

**En archivo Moneda.cpp tendríamos:**

```
#include "Moneda.h"

Moneda::Moneda() {
  miles=0;
}
  Constructores : 3/3
Moneda::Moneda(uint diezMil, uint cincoMil, uint dosMil, uint unMil){
  miles=10*diezMil+5*cincoMil+2*dosMil+unMil;
}

Moneda Moneda:: operator+(Moneda m) {   3/3
  Moneda r;
  r.miles = miles+m.miles;
  return r;
}

Moneda Moneda::operator-(Moneda m){     3/3
  Moneda r;
  r.miles = miles-m.miles;
  return r;
}

ostream & operator<< (ostream &os, const Moneda &m){   4/4
  int billetes=m.miles/10;
  int saldo=m.miles%10;
  os << "Billetes de diez mil=" << billetes << endl;
  billetes = saldo/5;
  saldo = saldo%5;
  os << "Billetes de cinco mil=" << billetes << endl;
  billetes = saldo/2;
  saldo = saldo%2;
  os << "Billetes de dos mil=" << billetes << endl;
  os << "Billetes de un mil=" << saldo << endl;
  return os;
}

Moneda operator* (int i, const Moneda &m)     2/2
{
  Moneda r;
  r.miles=i*m.miles;
  return r;
}

Un ejemplo de main sería (no se pedía, se probó en aragorn)

#include "Moneda.h"
using namespace std;

int main (void) {
  Moneda x, y(3, 2, 2, 2), z(2, 1, 3, 4);
  x = y+z;
  cout << x;
  x=2*z;
  cout << x;
  x= 2*y-z;
  cout << x;  // arroja el mínimo numero de billetes que representa el valor,
                //  cantidades negativas ponerlas entre paréntesis.
}
```

3.

a) ¿Bajo qué situación o circunstancia un programador debería considerar crear un Template en C++? 6/6
*Cuando una función o clase se repite con igual código para tipos de datos distintos -o en otras palabras posee el mismo patrón.*

b) ¿Qué mecanismo en Java equivale o permite atender la misma necesidad que las Temples de C++? 6/6
*Esta capacidad es lograda haciendo uso de interfaces. El patrón se programa haciendo uso de tipos genéricos (las interfaces) y luego estas interfaces deben ser implementadas por los tipos para los cuales el patrón tiene sentido. También juega un rol el hecho que todas las clases heredan de Object, las clases que agrupan datos como ArrayList hacen uso de esto.*

c) ¿Dé una ventaja o una desventaja de solución vía Templates en C++ respecto a la solución de Java? 6/6
*Una desventaja de las templates respecto a la solución Java es que el código resultante con template crece por cada instanciación de la template para tipos de datos específicos.*

d) ¿Cómo portaría usted a C++ el método de alguna clase Java descrito a continuación? ¿Habría algún requerimiento para la clase de key de su código C++?.

```
static int find (Object [ ] a , Object key)
{
    int i;
    for (i=0; i < a.length; i++)
       if (a[i].equals(key) return i;   // encontrado
    return -1;         // no exitoso
}
```

5/5

```
template<class T>
static int find ( T a[], int length, T key)
{
  int i;
  for (i=0; i < length; i++)
    if (a[i] == key) return i;   // encontrado
     return -1;         // no exitoso
}
```

2/2

***Sí, la clase de key debería tener definido el operador ==.***

---

4. Poniéndose en el caso que estamos en las fases iniciales de desarrollo de la aplicación del ejercicio 1 (no como applet, sino como la aplicación cuya implementación se muestra):
a) Escriba dos casos de uso para la aplicación mostrada en la pregunta 1. El primero describe el caso desde que el usuario ejecuta el programa hasta hacer el cuarto click. Considere como variante, el usuario hace un click fuera de la zona de la venta. En el segundo caso de uso, una vez que el usuario ha seguido el caso de uso previo, el usuario decide cubrir la ventana con otra aplicación y luego descubrirla, luego el usuario decide minimizar la aplicación y luego maximizarla.
b) Haga UML el diagrama de secuencia para el primer caso de uso desde que parte el programa hasta que el usuario presiona el segundo click en la ventana.

a) 7/7
Título: Ejecución primeros cuarto clicks
Actor: Usuario
Descripción: El usuario inicia la aplicación y generan los primeros trazos.
Flujo Principal:
1.- El usuario desde la línea de comandos o consola ejecuta la aplicación.
2.- El usuario mueve el mouse a la zona interior de la ventana.
3.- El usuario presiona una vez del mouse.
4.- El usuario mueve el mouse a otro punto de la ventana y lo vuelve a presionar.
5.- La aplicación muestra una línea entre ambos puntos indicados con clicks del mouse.
6.- Se repite 4 y 5 dos veces más.
Variante: 4/4
4A1.- El usuario mueve el mouse fuera de la ventana y lo presiona.
5A1.- El sistema no muestra línea pues ese evento no es notificado a la aplicación.
6A1.- continua en paso 6 (ó pasos 4 y 5).

Título: Respuesta a traslapo y minimización 4/4
Actor: Usuario
Descripción: Una vez dibujado algunos trazos, el usuario cubre la aplicación, luego la minimiza y maximiza.
Flujo Principal:
1.- Se sigue el caso “Ejecución primeros cuarto clicks”.
2.- El usuario cubre y descubre la aplicación con otra ventana.
3.- Una vez descubierta la aplicación muestra los trazos previamente hechos.
4.- El usuario minimiza y maximiza la aplicación.
5.- La minimización y maximización siguen comportamiento de toda ventana y una vez maximizado la aplicación muestra los trazos previamente hechos.

b) 10/10