Certamen Final

1. Se tiene una aplicación Java que muestra dos botones. Cuando presionamos el botón de la izquierda, incrementa el número mostrado en el botón de la derecha. Cuando presionamos el botón de la derecha, incrementa el número de la izquierda.

El código que implementa tal aplicación es:


```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class BotonesCruzados extends JFrame {
  public BotonesCruzados(String s)  {
    super(s);
    botonA= new JButton("0");
    botonB= new JButton("0");
    botonA.addActionListener( new ActionListener()   {
       public void actionPerformed(ActionEvent event)  {
        String texto=botonB.getText();
        texto=Integer.toString(Integer.parseInt(texto)+1);
        botonB.setText(texto);
       }
      });
    botonB.addActionListener( new ActionListener()   {
       public void actionPerformed(ActionEvent event)   {
        String texto=botonA.getText();
        texto = Integer.toString(Integer.parseInt(texto)+1);
        botonA.setText(texto);
       }
      });
    JPanel panel= new JPanel();
    panel.add(botonA);
    panel.add(botonB);
    getContentPane().add(panel);
    pack();
  }
  public static void main (String[ ] args) {
    BotonesCruzados botonesFrame = new BotonesCruzados("Pregunta_1");
    botonesFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    botonesFrame.show();
  }
  private JButton botonA;
  private JButton botonB;
}
```

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 valor inicial con que parte cada uno de los contadores.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class BotonesCruzadosApplet extends JApplet {
public void init() {
   botonA= new JButton(getParameter("contadorIzq"));
```

```
    botonB= new JButton(getParameter("contadorDer"));
    botonA.addActionListener( new ActionListener()  {
      public void actionPerformed(ActionEvent event) {
       String texto=botonB.getText();
       texto=Integer.toString(Integer.parseInt(texto)+1);
       botonB.setText(texto);
      }
     });
    botonB.addActionListener( new ActionListener()  {
      public void actionPerformed(ActionEvent event)  {
       String texto=botonA.getText();
       texto = Integer.toString(Integer.parseInt(texto)+1);
       botonA.setText(texto);
      }
     });
    JPanel panel= new JPanel();
    panel.add(botonA);
    panel.add(botonB);
    getContentPane().add(panel);
  }
  private JButton botonA;
  private JButton botonB;
}
```

b) Muestre el contenido de un archivo html que use su applet con contadores partiendo ambos en 5.

```
<html>
<title>Botones Cruzados</title>
<body>
<applet code="BotonesCruzadosApplet.class"
  width="150" height="50">
<param name="contadorIzq" value="5"/>
<param name="contadorDer" value="5"/>
</applet>
</body>
</html>
```

2. Poniéndose en el caso que estamos en las fases iniciales de desarrollo de la aplicación del ejercicio 1:

a) Escriba dos casos de uso para el applet de la pregunta 1. El primero es: Uso del applet. Este caso de uso describe lo que un usuario de su applet debe hacer para incorporarla en una página web. El caso de uso termina describiendo lo que el usuario ve -relacionado con el applet- en la página web (sin que aún haya interactuado con ella). El segundo caso de uso describe lo que ocurre cuando el usuario presiona y libera el botón izquierdo.

Nombre: Uso del applet

Actor: Desarrollador o Persona hace uso del applet, visitante a sitio web

Descripción: Un usuario del applet la incorpora a una página web y luego un visitante la despliega usando un navegador.

Flujo principal:

1.- En el lugar del archivo html donde el usuario desea ver el applet, él pone

```
<applet code="BotonesCruzadosApplet.class"
  width="150" height="50">
<param name="contadorIzq" value="5"/>
<param name="contadorDer" value="6"/>
</applet>
```

el valor asociado a contador izquierdo y derecho corresponde al valor inicial para cada contador.

2.- El visitante al sitio web y haciendo uso de un navegador abre el archivo html generado previamente.

3.- El navegador despliega dos botones en la zona correspondiente al applet. Cada botón tiene como rótulo o etiqueta el número especificado para los parámetros contadorIzq y contadorDer.

Nombre: Presionar y liberar botón izquierdo

Actor: Visitante a sitio web

Descripción: El visitante actúa sobre el applet presionando y liberando el botón izquierdo.

Flujo principal:

1.- El visitante usando un navegador abre una página html que contiene el applet.

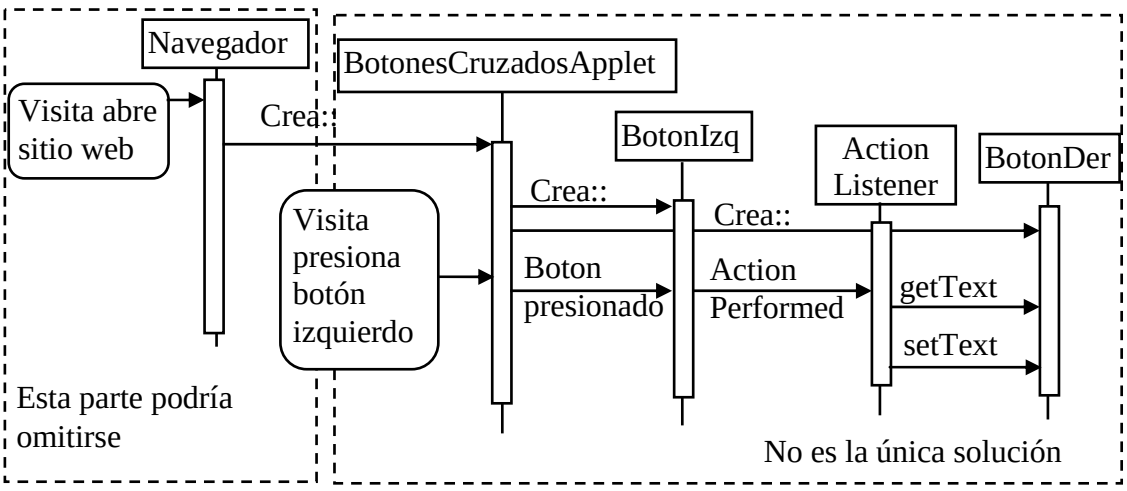
2.- El navegador ejecuta el applet mostrando dos botones.

3.- El visitante presiona y libera el botón izquierdo del applet.

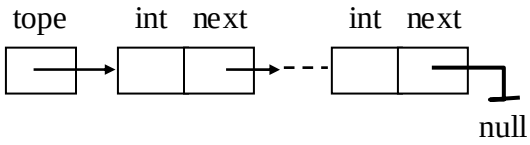
4.- El applet incrementa y muestra el número mostrado como rótulo en el botón derecho.

b) Haga un diagrama de secuencia UML que muestre el segundo caso de uso (presionar y liberar botón izquierdo)

El diagrama de secuencia UML muestra la interacción entre los objetos que intervienen en el caso de uso:



3. En C++ proponga una clase para StackDinamico de enteros. Un stack es una estructura de datos que ofrece las operaciones en este caso: push(…) para insertar un nuevo entero en él; pop (…) para sacar el elemento del tope del stack; top (..) que retorna una referencia al elemento del tope sin sacarlo del stack; y estaVacio(…) que retorna si el stack está o no vacío. Se pide que el stack sea implementado como una estructura dinámica, es decir sus elementos sean almacenados en una lista del tipo:



Se pide: a) declaración de la clase (archivo .h) b) implementación del método estaVacio y el destructor.

a) StackDinamico.h

```
#include <iostream>
using namespace std;

class Nodo {
public:
  int dato;
  Nodo * next;
};

class StackDinamico {
public:
  StackDinamico();
  StackDinamico(const StackDinamico & sd);
  void push(int a);
  int pop();
```

```
  int top() const;
  bool estaVacio() const;
  const StackDinamico & operator =(const StackDinamico & sd);
  ~StackDinamico();
private:
  Nodo * tope;
};
```

b) StackDinamico.cpp

```
bool StackDinamico::estaVacio() const {
  return (tope == NULL);
}

StackDinamico::~StackDinamico() {
  while(tope != NULL) {
    Nodo * tmp = tope->next;
    delete tope;
    tope = tmp;
  }
}
```

4. Se tiene el siguiente código:

```
#include <iostream>
using namespace std;
class P4 {
public:
 P4() {  a=5; }
 int & getA() { return a;}
private:
  int a;
};
int main () {
  P4 p4;
  cout << p4.getA()++ <<" "<< p4.getA()<< endl;
  p4.getA() = 20;
  cout << p4.getA()++ <<" "<< p4.getA() << endl;
}
```

a) Indique si tiene errores de compilación. Justifique.

No hay errores de compilación. Se hace uso adecuado de referencias en C++.

b) Si no hubiera errores, muestre y explique lo que haría el programa. Si hay errores, sugiera una corrección a su gusto para que el programa haga algo similar a lo insinuado en el código.

El programa mostrará:

5 6
20 21
El método getA() retorna una referencia al atributo a con lo cual el programa principal puede cambiar su valor. En este caso no requerimos que a sea visible. La versión C de esta situación es retornar un puntero a ***a,*** como:
```
int * getA() { return &a;}
```
y en el main hacer algo como *p4.getA()=20;

c) ¿Cambiaría o no su respuesta a) y b) si el atributo de la clase P4 fuera protected ~~private~~? Justifique.
Ambas respuestas no cambian. El calificativo protected es menos restrictivo que el private y sólo modifica la situación para subclases de P4, que no es el caso de esta pregunta.

d) ¿Cambiaría o no su respuesta a) y b) si el método getA() fuera virtual? Justifique.
Ambas respuestas no cambian. El calificativo virtual modifica la situación para subclases de P4 que deseen redefinir este método, que no es el caso de esta pregunta.

e) ¿Cambiaría o no su respuesta a) y b) si el método de la clase fuera declarado como:
`int & getA() const { return a;}? Justifique.`
La respuesta cambia. La idea detrás de declarar un método const es que éste no permitirá que los atributos del objeto sean cambiados. Esto no se cumple si el método retorna una referencia a un atributo.

f)¿Cambiaría o no su respuesta a) y b) si el método de la clase fuera declarado como:
`const int & getA() { return a;}? Justifique.`
La respuesta también cambia en este caso. Habría un error de compilación en este caso. Se dice que la referencia retornada es constante, pero en su uso del main se intenta cambiar su valor al incrementarlo y asignarle otro valor. Esta definición es incompatible con lo que se desea hacer en el main.