Auxiliar Extra Control 2 (Solución) – Profesor Nelson Baloian
Jueves 30 de abril de 2009

```
//Pregunta 1
//clase Caballo: 2.0 ptos
//encabezamiento y constructor: 0.5 ptos
class Caballo extends Pieza{
public Caballo(int x,int y){super(x,y);}
//método mover: 1.5 ptos
public boolean mover(int x,int y){
  if( (1<=x && x<=8) && (1<=y && y <=8)
  &&(Math.abs(x-obtenerFila())==2 && Math.abs(y-obtenerColumna())==1)
  ||(Math.abs(x-obtenerFila())==1 && Math.abs(y-obtenerColumna())==2)))
    return super.mover(x,y);
  else
    return false;
}
//clase Torre: 2.0 ptos
//encabezamiento y constructor: 0.5 ptos
class Torre extends Pieza{    //0.1
public Torre(int x,int y){ super(x,y);}
//método mover: 1.5 ptos
public boolean mover(int x,int y){
  if( (1<=x && x<=8) && (1<=y && y <=8)
  && (( x == obtenerFila() && y != obtenerColumna() )
  ||( x != obtenerFila() &&  y == obtenerColumna() ) )
    return super.mover(x,y);
  else
    return false;
}
//método comer: 2.0 ptos
boolean comer(Pieza X, Pieza Y){
  int i=Y.obtenerFila(), j=Y.obtenerColumna();
  return X.mover(i,j);
}

//Pregunta 2
//a)
//clase Rectangulo: 1.5
class Rectangulo implements I{
private int a,b;
public String escribir(){
  return a+" "+b;
}
public int perimetro(){
  return 2*(a+b);
}
public void leer(String x){
  int i=x.indexOf(" ",1);
  a=Integer.parseInt(x.substring(0,i));
  b=Integer.parseInt(x.substring(i+1));
}
}
//clase Triangulo: 1.5
class Triangulo implements I{
private int a,b,c;
```

```
public String escribir(){
  return a+" "+b+" "+c;
}
public int perimetro(){
  return a+b+c;
}
public void leer(String x){
  int i=x.indexOf(" ",1);
  a=Integer.parseInt(x.substring(0,i));
  int j=x.indexOf(" ",i+1);
  b=Integer.parseInt(x.substring(i,j));
  c=Integer.parseInt(x.substring(j+1));
}
}

//b)
//declaraciones iniciales: 0.3
I max=new Rectangulo();
max.leer("0 0");
//leer hasta fin de archivo: 0.2
BufferedReader a=new BufferedReader(new FileReader("figura.txt"));
String linea;
while((linea=a.readLine())!=null)
{
  //crear objeto: 1.0
  I f;
  if(linea.charAt(0)=='R')
     f=new Rectangulo();
  else
     f=new Triangulo();   /
  //leer lados: 0.5
  f.leer(linea.substring(1));
  //mantener y escribir el mayor: 1.0
  if(f.perimetro()>max.perimetro())
    max=f;
}
System.out.println(max.escribir());

//Pregunta 3
import java.awt.*;
import java.awt.event.*;
class Fractal extends Frame implements ActionListener{
static public void main(String[]args){
      new Fractal().setVisible(true);
}
//declaraciones: 0.5
private Canvas cv=new Canvas();
private TextField
  profundidad=new TextField("10"),
  grados=new TextField("30");
private Button
  dibujar=new Button("dibujar"),
  cerrar=new Button("cerrar");
final private int W = 200;
//constructor: 2.0
public Fractal(){
  //diagramar panel norte: 0.5
```

```
  Panel p1=new Panel();//0.1
  p1.setLayout(new GridLayout(2,2));
  p1.add(new Label("profundidad:"));p1.add(profundidad);
  p1.add(new Label("angulo:")); p1.add(grados);
  //diagramar panel sur: 0.5
  Panel p2=new Panel();
  p2.setLayout(new GridLayout(1,2));
  p2.add(dibujar);
  p2.add(cerrar);
  //diagramar ventana: 0.5
  setSize(W,3*W/2); //opcional
  cv.setSize(W,W);
  setLayout(new BorderLayout());
  add("North",p1);
  add("Center",cv);
  add("South",p2);
  //activar escuchadores: 0.5
  dibujar.addActionListener(this);
  cerrar.addActionListener(this);
}
//actionPerformed: 1.5
public void actionPerformed(ActionEvent x){
  if(x.getSource()==cerrar)
    System.exit(0);
int n=Integer.parseInt(profundidad.getText());
double alfa = Double.parseDouble(grados.getText());
  //invocación a dibujar arbol: 1.0
  dibujarArbol(
    n,
    W/n,
    90,
    W/2,
    W,
    cv.getGraphics());
}
//dibujarArbol: 2.0
public void dibujarArbol(int profundidad,int largo, double angulo,
    int x,int y,Graphics g){
  //caso base: 0.2
  if(profundidad==0)
    return;
  //dibujar tronco: 0.8
  double radianes=Math.PI*angulo/180;
  int xf=(int)(x-largo*Math.cos(radianes));
  int yf=(int)(y-largo*Math.sin(radianes));
  g.drawLine(x,y,xf,yf);
  //llamadas recursivas para dibujar 2 ramas: 1.0
  int alfa=Integer.parseInt(grados.getText());
  dibujarArbol(profundidad-1, (int)(largo*0.67), angulo+alfa/2, xf,yf,g);
  dibujarArbol(profundidad-1, (int)(largo*0.67), angulo-alfa/2, xf,yf,g);
}
}
```