Examen: Programación II

Grado Multimedia
DLSI – Universidad de Alicante
Julio 2011

Aclaraciones:

- Duración del examen: 3 horas.
- Realiza cada ejercicio en un folio separado.
- Pon apellidos, nombre y DNI en todas las hojas.
- La publicación de notas del examen se realizará a través del Campus Virtual aproximadamente una semana después de la realización del mismo.
- Junto con las notas provisionales se publicará la fecha de revisión del examen.
- La publicación de notas definitivas se realizará dos días después de la revisión del examen.

Preguntas

Pregunta 1 (3 p.):

Un robot industrial tiene la posibilidad de acoplarse distintos tipos de herramientas. Todas las herramientas tienen dos operaciones en común: `Abrir` y `Cerrar` y en principio el robot puede acoplarse: `Pinzas`, `Tenazas` y `Alicates`.

Teniendo en cuenta que a priori el robot no sabe qué herramienta se va a acoplar, se pide:

1. Implementa el código java, haciendo uso de una `interface`, correspondiente a las clases de las herramientas.[1]
2. Implementa la clase `Robot` de manera que:
   - Contenga la herramienta que se va a acoplar de manera que sólo sea accesible por el robot.
   - El constructor de manera que crea la herramienta que se acopla el robot. El tipo de herramienta será uno u otro en función del parámetro, que será de tipo entero. Si el parámetro es igual a `1` crea unas `Pinzas`, si el parámetro es igual a `2` crea unas `Tenazas` y en cualquier otro caso crea unos `Alicates`.
   - Los métodos `Abre` y `Cierra` que hacen que el robot abra y cierre la herramienta, respectivamente.

[1] Las operaciones de `Abrir` y `Cerrar` se limitan a escribir un mensaje en pantalla.

Pregunta 2 (2 p.):

Dado el siguiente código:

```
class Super{
 protected double x;
 public Super(double v) { x=v; }
 public double Suma(double incr) { x=x+incr; return x;}
 public double Resta(double decr) { x=x-decr; return x;}
}
class Sub extends Super{
 public Sub(double v) { super(v); }
 public int Suma(int incr) {
  int devolver;
  if (incr<0)
   devolver=(int) x;
  else
   devolver=(int)super.Suma(incr);
  return devolver;
 }
 public double Resta(double decr){
  double devolver=0;
  if((decr>0)&&(x>=decr)){
   x=x-decr; devolver=x;
  }
 return devolver;
}
}
class Uso{
 public static void main(String[] args){
  Super asaber=new Sub(2.5);
  System.out.println(asaber.Suma(-2));
  System.out.println(asaber.Resta(10));
}
}
```

1. Indica que salida produce y justifica tu respuesta.
2. ¿Qué tipo de sobrecarga se está utilizando con el método `Suma`?

Pregunta 3 (1 p.):

Indica cómo podemos distinguir una variable de instancia de una superclase de una variable de instancia que se llame igual en una de sus subclases. Pon un ejemplo.

Pregunta 4 (2 p.):

Dado el siguiente código:

```
public class Programa{
 int x = 1;
 static int y = 2;

 public static int m1(int x){
  x = 4;
  return x*x;
 }
 public void m2(){
  --x;
 }
 public int m3(){
  int x = 5;
  this.x = x * this.x;
  return x;
 }
 public static void m4(int y){
  this.y=y;
 }
 void metodo(){
   int x = 6;
   m3();
   System.out.println("x = " + x);
   System.out.println("x = " + this.x);
   m2();
   x = m1(this.x);
   System.out.println("x = " + x);
   System.out.println("x = " + this.x);
   m3();
   System.out.println("x = " + this.x);
   System.out.println("x = " + m3());
 }
 public static void main(String[] args){
   (new Programa()).metodo();
   m4(10);
   System.out.println("y = "+ y);
 }
}
```

1. Indica qué error contiene el código y cómo solucionarlo.
2. Indica que salida mostraría el código correcto.

Pregunta 5 (2 p.):

Implementa una clase `Indices` que busque los valores `máximo` y `mínimo` en un array. Para ello se necesita:

- Un método de clase llamado `IndiceValorMinimo` que reciba como parámetro un array de enteros y devuelva un entero con la posición del array que contiene la primera aparición del valor mínimo de éste.[2]
- Un método de clase llamado `IndiceValorMaximo` que reciba como parámetro un array de enteros y devuelva un entero con la posición del array que contiene la primera aparición del valor máximo de éste.
- En ambos casos si el array no existe el método devolverá el valor `-1`.

Para probar su funcionamiento implementa una clase `Prueba` con un método `main` que contenga una serie de instrucciones que hagan uso de los métodos definidos.

[2]Se debe tener en cuenta que es un array de enteros y por tanto puede contener valores negativos.