Departamento de Enxeñería Telemática
Universidade de Vigo

16 de diciembre de 2013

PROGRAMACIÓN II **(Duración del examen: 2 horas)**

NOMBRE:

Cuestión 1. **(0.5 puntos)**

Indique si las siguientes afirmaciones sobre características de Java son verdaderas o falsas. **NOTA: Fallar alguna de las afirmaciones supone invalidar la cuestión entera.**

1. Una clase abstracta no puede incluir métodos concretos _____
2. Al sobrecargar un método se puede modificar el tipo pero no el número de argumentos de entrada _____
3. Las clases abstractas no tienen constructor porque no se pueden instanciar _____
4. En una interfaz podemos definir métodos protegidos (*protected*) abstractos, pero nunca métodos protegidos (*protected*) concretos _____
5. Una interfaz puede incluir tanto atributos como métodos _____

Cuestión 2. **(0.5 puntos)**

Dado el siguiente código:

```
public class Clase1{
	public int atributo;
	public Clase1(int a){
		atributo=a;
	}
}
public class Clase2{
	public double atributo;
	public Clase2(double a){
		atributo=a;
	}
}
```

¿Cuál es el resultado de ejecutar el siguiente código Java?

```
public class Test {
  public static void main (String[] args){
	int entero = 25;
	Clase1 obj1 = new Clase1(1);
	Clase2 obj2 = new Clase2(2.2);
	metodo (entero,obj1,obj2);
	System.out.println("El valor del entero es "+entero);
	System.out.println("El atributo de obj1 es "+obj1.atributo);
	System.out.println("El atributo de obj2 es "+obj2.atributo);
  }
  public static void metodo (int e, Clase1 c1, Clase2 c2){
	e=100;
	c1.atributo=5;
	c2=new Clase2(4.7);
	System.out.println("El atributo de c2 es "+c2.atributo);
  }
}
```

Respuesta:

Cuestión 3. (0.5 puntos)

¿Es correcto el fragmento de código Java mostrado a continuación? En caso de error, describir una posible solución para que el programa se ejecute correctamente y muestre por pantalla los valores de los atributos "*entero*" y "*cadena*".

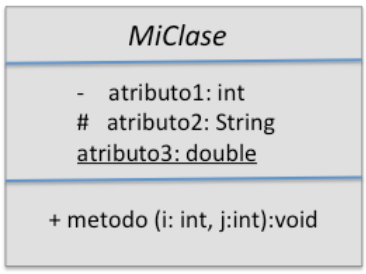
```
package p1;
public class ClaseA{
	public int entero;
	String cadena;

	public ClaseA (){
		entero = 1;
		cadena = "Hola mundo!";
	}
}
```

```
package p2;
import p1.ClaseA;
public class ClaseB extends ClaseA{
	public static void main (String[] args){
		ClaseB c = new ClaseB();
		System.out.println("El entero vale " + c.entero + " y la cadena " +
c.cadena);
	}
}
```

Respuesta:

Cuestión 4. (0.5 puntos)

Escribir el código Java correspondiente al diagrama UML mostrado en la figura.



Respuesta:

Cuestión 5. (0.5 puntos)

¿Cuál es el resultado de ejecutar el siguiente código Java?

```
public class Test {
	public static void main (String[] args){
		try{
			divide(5,0);
			System.out.println("Mensaje 1");
		}
		catch(java.lang.ArithmeticException ex){
			System.out.println("Mensaje 2");
		}
		catch(java.lang.Exception ex){
			System.out.println("Mensaje 3");
		}
		finally{
			System.out.println("Mensaje 4");
		}
		System.out.println("Mensaje 5");
	}
	public static void divide (int n, int d){
		try{
			System.out.println("Resultado: "+n/d);
		}catch(java.lang.NullPointerException ex){
			System.out.println("Mensaje 6");
		}
		finally{
			System.out.println("Mensaje 7");
		}
		System.out.println("Mensaje 8");
	}
}
```

Respuesta:

Problema 1. (1.5 puntos)

Este problema gira en torno a un zoo con las características descritas a continuación:

- En el recinto hay únicamente cinco especies de animales: tucanes y halcones (en el grupo de las aves) y osos, koalas y jirafas (en el grupo de los mamíferos).
- Todos los animales pueden ser visitados en el horario acordado por los gestores del zoo (disponible a través del método `String getHorarioVisita()`) salvo los koalas, que están actualmente en tratamiento veterinario tras sufrir una sobredosis de eucalipto.
- El horario de visita de los osos y las jirafas es "*Lu-Vi de 10 a 17*", el de los tucanes es "*Sa de 12 a 16*" y el de los halcones "*Do de 12 a 16*".
- Además de un identificador único (ID) y el peso del bicho, la ficha de algunos animales registra parámetros específicos: fecha de hibernación (para los osos), kilogramos de eucalipto ingeridos (para los koalas) y color del plumaje (para las aves).

Se pide:

1. **(0.5 puntos)** Escribir el código Java para modelar el zoo, sabiendo que: (i) **deben utilizarse necesariamente interfaces** en el diseño de la solución, (ii) deben especificarse los atributos de cada clase (concreta o abstracta), los cuales tendrán que ser todos **PRIVADOS**, y (iii) deben definirse sus métodos (se pueden omitir los constructores en este aparatado).

Respuesta:

2. **(0.25 puntos)** Añadir el código necesario en las clases definidas en el apartado anterior para poder crear un objeto que represente un tucán con ID “222”, 1.5 kg de peso y plumas de color “negro” mediante la siguiente secuencia:

Tucanes tucan = new Tucanes (“222”,1.5,”negro”);

Respuesta:

3. **(0.25 puntos)** Represente el diagrama UML correspondiente a la jerarquía de clases e interfaces diseñada en los apartados anteriores. **NOTA: En caso de ser necesario, escribir en letra mayúscula lo que deba ser representado en cursiva/itálica.**

Respuesta:

4. **(0.5 puntos)** Considerando los atributos **privados** de las clases definidas en el primer apartado y asumiendo que existe un método `getZoo()` que permite recuperar **todos** los animales del zoo, escribir el código Java necesario para mostrar por pantalla el horario de visita de los animales almacenados en el array `Animal[] todosAnimalesZoo` incluido a continuación. En el caso de los osos se imprimirá también el ID y la fecha de hibernación.

```
public class Test {
  public static void main (String[] args){
       Animales[] todosAnimalesZoo = getZoo();
       for (int i=0; i<zoo.length; i++){









       }
}
}
```

Problema 2. (1 punto)

Dada la clase Circulo mostrada a continuación:

```
public class Circulo {
  private double radio;
  public Circulo (double r){
      radio = r;
  }
  public boolean equals (Circulo otro){
    return this==otro;
  }
}
```

1. **(0.25 puntos)** Indicar cuál es el resultado de ejecutar el siguiente fragmento de código Java:

```
public class Test {
  public static void main (String[] args){
     Circulo c = new Circulo (2.2);
     System.out.println("¿Círculos iguales?"+c.equals(new Circulo(2.2)));
  }
}
```

Respuesta:

2. **(0.75 puntos)** Modificar la clase Circulo de forma que el resultado de ejecutar el siguiente código sea: **Los círculos comparados son iguales!**.

```
public class Test{
   public static void main (String[] args){
     Object c= new Circulo (2.2);
     if (c.equals (new Circulo (2.2))
         System.out.println("Los círculos comparados son iguales!");
   }
}
```

Respuesta:

```
public class Circulo {
  private double radio;
  public Circulo (double r){
      radio = r;
  }









}
```