

Prueba 3 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta en esta misma hoja.

Apellidos:

Nombre:

Apartado 1. (4 puntos)

Completa el siguiente programa, para que su ejecución resulte en la siguiente salida:

```
{Bolt, Usain (1986)=9.58, Powel, Asafa (1982)=9.72, Thompson, Richard (1985)=9.89, Thompson, Richard (1995)=10.92}
```

```
import ... // Complétalo
...

class Atleta .....{ // Complétalo
    private String nombre, apellidos;
    private int fnacimiento;

    public Atleta(String n, String a, int fn) {
        this.nombre = n;
        this.apellidos = a;
        this.fnacimiento = fn;
    }
    // Completa el código que falta a continuación.....

}

public class Colecciones {
    public static void main(String[] args) {
        ..... marcaPersonal = new .....(); // Complétalo
        // Nombre, Apellidos y fecha de nacimiento
        marcaPersonal.put( new Atleta("Usain", "Bolt", 1986), 9.58);
        marcaPersonal.put( new Atleta("Richard", "Thompson", 1995), 10.92);
        marcaPersonal.put( new Atleta("Richard", "Thompson", 1985), 9.89);
        marcaPersonal.put( new Atleta("Asafa", "Powel", 1982), 9.72);

        System.out.println(marcaPersonal);
    }
}
```


Prueba 3 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta en esta misma hoja.

Apellidos:

Nombre:

Apartado 2. (3.5 puntos)

Se quiere construir una clase que filtre los elementos de una Lista (java.util.List) quedándose con aquellos que cumplen cierta condición. La condición se da mediante un objeto que implementa la interfaz genérica Filtro.

Completa el siguiente programa, para que su ejecución resulte en la siguiente salida:

Lista filtrada: [Java, lenguaje, favorito]

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

interface Filtro<T> {
    public boolean eval(T x);
}
// Definimos una clase filtro que seleccione los Strings de longitud distinta de 2.
class BorraPalabras2Letras .....{ // Complétalo
    // Completa el código.....
}

class ListaFiltrada .....{
    // Completa el código.....
}

}

public class Genericos {

    public static void main(String[] args) {
        List<String> lst = Arrays.asList("Java", "es", "mi", "lenguaje", "favorito");
        ListaFiltrada<String> f =
            new ListaFiltrada<String>(lst, new BorraPalabras2Letras());
        System.out.println("Lista filtrada: "+f);
    }
}
```


Prueba 3 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta en esta misma hoja.

Apellidos:

Nombre:

Apartado 3. (2.5 puntos)

¿Qué salida produce la ejecución del siguiente programa?

```
class ExcepcionNumerica extends Exception {
    public ExcepcionNumerica () { System.out.println("Creada Excepcion Numerica"); }
}
class ExcepcionFactorial extends ExcepcionNumerica {
    public ExcepcionFactorial (String s) { System.out.println("Excepcion Factorial: " + s); }
}

public class ManejandoExcepciones {
    private static void imprimeFactoriales(int[] nums) {
        int i = 0;
        try {
            for (int n : nums) {
                System.out.println("factorial(" + n + ") = "+ factorial(n));
                i++;
            }
        } catch (ExcepcionFactorial e){
            System.out.println("en factorial(" + nums[i] + ").");
        } catch (ExcepcionNumerica e) {
            System.out.println("Error en factorial(" + nums[i] + ").");
        } finally {
            System.out.println("Se han impreso " + i + " factoriales.");
        }
    }

    private static int factorial(int n) throws ExcepcionNumerica {
        if (n < 0) throw new ExcepcionNumerica();
        if (n > 12) throw new ExcepcionFactorial("resultado demasiado grande para int");
        if (n == 0) return 1;
        return n * factorial(n-1);
    }

    public static void main(String[] args) throws Exception {
        int[] v = {2, -5, 0, 16};
        int[] w = {0, 14, -3, -9};
        imprimeFactoriales(v);
        imprimeFactoriales(w);
    }
}
```

Salida Esperada:

