

Trabajo Práctico Nº 2

Entrega Individual: 6 de abril a arielmiglio@gmail.com

Consideraciones para la entrega

La entrega se realizará por e-mail y deberá estar compuesta por el proyecto de eclipse comprimido y un documento de texto (doc o similar) con las resoluciones de aquellos ejercicios que sean teóricos.

El nombre del proyecto debe tener la forma trabajo-practico2-<nombre y apellido del alumno>.

Por ejemplo: trabajo-practico2-Simpson-Homero

Por último, las clases de cada ejercicio deben estar en paquetes con el nombre del mismo. Si hubiese más de un inciso, usar notación puntual para el nombre del paquete, ejemplo: ejercicio1.incisoA y ejercicio1.incisoB.

Ejercitación

- 1) Cierta banco posee dos tipos de cuentas bancarias, Caja de Ahorro y Cuenta corriente. Ambas poseen un número de cuenta, un saldo y se les pueden realizar depósitos y extracciones. Sin embargo en la Caja de Ahorro no se puede tener saldo negativo, mientras que sobre las Cuentas Corrientes se pueden realizar hasta 30 extracciones sin importar el importe.
 - a.- Además, cada cuenta bancaria posee un titular del que se conoce nombre y DNI.
 - b.- Implementar las clases necesarias, con sus atributos y métodos
- 2) Realice un programa java que cree un arreglo con 50 números enteros con valores aleatorios entre 0 y 100.
Nota: puede ayudarse con el random de la clase Math.
- 3) Utilizando lo programado en el ejercicio anterior, realice un programa que a partir del arreglo con los cincuenta números aleatorios imprima en pantalla la suma de los pares, la suma de los impares, la cantidad de múltiplos de 2 y la cantidad de múltiplos de 7.
- 4) Dado el siguiente código indique que valores retornan o si dan error por qué se produce, las expresiones desde la a) hasta la j).

```
String a = "abc";  
String s = a;  
String t;
```

- a) s.length();
- b) t.length();
- c) 1 + a;
- d) a.toUpperCase();
- e) "Libertad".indexOf("r");
- f) "Universidad".lastIndexOf('i');
- g) "Quilmes".substring(2,4);
- h) (a.length() + a).startsWith("a");

- i) `s == a;`
- j) `a.substring(1,3).equals("bc")`

- 5) Un pangrama es una frase que contiene todas las letras del abecedario. Son utilizados en pruebas de impresión para ver en una frase corta como quedan las diferentes letras de una tipografía particular. Programe un pangramador, el cual debe recibir desde línea de comandos una frase y como resultado debe decir cuantas letras del abecedario faltaron y cuantas palabras contiene dicha frase. Por ejemplo “Queda gazpacho, fibra, látex, jamón, kiwi y viñas” es un pangrama.
- 6) Cree una clase que se encuentre en paquete `model`, otra clase que se encuentre en el paquete `model.gui` y otra que se encuentre en el paquete `model.stack`. Compílelas utilizando la forma que cree conveniente.
- a. ¿Cómo están organizadas en el sistema de archivos?
 - b. ¿Encuentra alguna relación entre el nombre del paquete y la ubicación de los archivos fuentes de las clases y los archivos compilados?
- 7) Teniendo en cuenta la siguiente definición de clases :

```
package model;

public abstract class Empleado {

    public abstract float sueldo();
}
```

- a. Teniendo en cuenta la siguiente implementación. ¿Qué es necesario agregar para que pueda compilar? Piense en todas las alternativas posibles.

```
package model;

public class EmpleadoTemporal extends Empleado{

}
```

- 8) Diseñe e implemente la clase `Point` (Punto). La clase debe tener el siguiente comportamiento.
- a. Debe ser posible crearse indicando como referencia los valores `x` e `y`, también debe ser posible crear un punto directamente sin enviarles parámetros, en estos casos el punto debe crearse en las coordenadas (0,0).
 - b. Debe ser posible mover un punto a otra posición.
 - c. Sumarse con otro punto y como resultado obtener un nuevo punto con las valores de `x` e `y` sumados.
- 9) Indicar que imprime el siguiente código

```
package modelo;
public class Persona{
    public void x(){
        System.out.println("modelo.Persona");
    }
}
```

```
package actores;
public class Persona {

    public void x(){
        System.out.println("actores.Persona");
    }
}
```

```
import modelo.*;
import actores.Persona;

public class main {

    public static void main (String [] args) {
        Persona persona = new Persona();
        persona.x();
    }
}
```