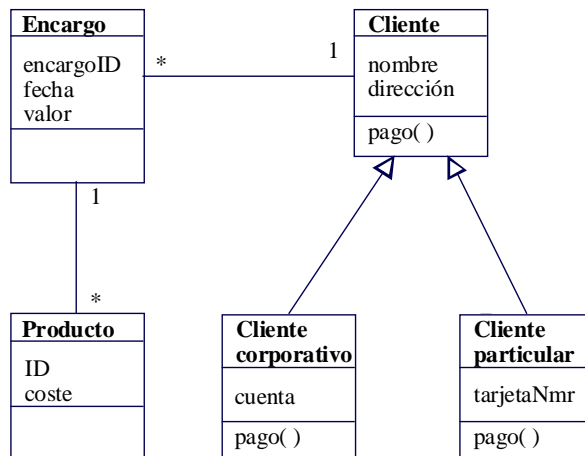


Ejercicios del tema 7: Clases y objetos

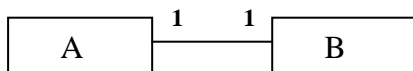
1. a) Diseñe e implemente en Java una clase para trabajar con triángulos isósceles. Para ello, defina las variables de ejemplar mínimas que requiera, proporcione métodos de consulta, un método constructor e implemente otros métodos para calcular el perímetro y el área de un triángulo.
- b) Implemente en Java un método que a partir de una matriz de triángulos devuelva el área del triángulo de mayor superficie

2. El siguiente diagrama de clases, incompleto, representa la estructura básica de un programa para recopilar los encargos de productos de una compañía.



Escriba una versión en Java de la clase Producto. Incluya métodos que permitan acceder a la información del producto.

3. Dada la siguiente relación de asociación entre las clases A y B. Indique como se implementaría



- a) Un objeto de A hereda de B
- b) Un método de B tiene como parámetro un objeto A
- c) Un atributo de ejemplar de B es un objeto A
- d) Ninguna de las anteriores es cierta

4. ¿Cuál es el resultado del siguiente programa?

```

class Ejercicio {
    public static void main(String [ ] args){
        Clase1 obj1=new Clase1();
        obj1.imprimir(24.3,5);
    }
}
class Clase1 {
    private double valor=9.8;
    private int x=7;
    public void imprimir(double valor, int x) {
        System.out.print(valor+" "+this.x);
    }
}
  
```

5. ¿Qué imprime el siguiente programa?

```

class Ejercicio {
    public static void main (String [] args) {
        Clase1 obj1= new Clase1(5,4);
        System.out.print(obj1.modificar(4)+" ");
        Clase1 obj2= new Clase1(5,4);
        System.out.print(obj2.modificar(5)+" ");
        obj2=obj1;
        System.out.println(obj2.modificar(5)+" ");
    } //fin main
}
class Clase1{
    int p1,p2;
    public Clase1 (int i, int j){
        p1=i;
        p2=j;
    }
    public int modificar(int i){
        p1=p1+i;
        p2=p2+i;
        System.out.print(p2+" ");
        return p1;
    }
}

```

6. Desarrolle un programa que sirva para evaluar el valor de un polinomio, cuyo grado y coeficientes se introducen por teclado, en un valor de abscisa determinado.

7. ¿Cuál es el error del siguiente programa?

```

class Cuenta {
    protected int numero_cuenta;
    protected double saldo;

    public Cuenta(int cuenta, double inicial) {
        numero_cuenta=cuenta;
        saldo = inicial;
    } // Fin constructor cuenta

    public void depositar(double cantidad) {
        saldo = saldo + cantidad;
    } // Fin método deposito

    public double saldo(){
        return saldo;
    } // Fin método saldo
} //fin clase Cuenta

class Banco {
    public static void main(String [] args) {
        //Se crea la cuenta
        Cuenta cuenta_1 = new Cuenta(123456, 2505.32);
        // Se consulta el saldo
        double total_cuenta=cuenta_1.saldo;
        System.out.println("Total actual en la cuenta: "
            +total_cuenta +" Euros");
    } //del main
} //de la clase Banco

```

8. Un teatro se caracteriza por su nombre y su dirección y en él se realizan 4 funciones al día. Cada función tiene un nombre y un precio. Realice el diseño de clases e indique qué métodos tendría cada clase, teniendo en cuenta que se pueda cambiar el nombre del teatro y el nombre y precio de la función. Implemente dichas clases.

9. a) ¿Qué imprime el siguiente programa? b) ¿Y si prop en Clase1 no la declaramos como static?

```
class Ejercicio {
    public static void main (String [] args) {
        Clase1 obj1= new Clase1(7);
        Clase1 obj2= new Clase1(9);
        obj2=obj1;
        System.out.print(obj1.valorprop()+" ");
        System.out.print(obj2.valorprop()+" ");
    } //fin main
}
class Clase1{
    static int prop=0;
    public Clase1 (int valor){
        prop=valor;
    }
    public int valorprop(){
        return prop;
    }
}
```

10. Dado el siguiente fragmento de código:

```
Ordenador sobremesa; Ordenador portatil;
sobremesa = new Ordenador();
sobremesa.precio(900);
portatil = new Ordenador();
portatil.precio(1100);
portatil = sobremesa;
sobremesa = null;
```

¿Qué afirmación es cierta?

- Al final tanto el objeto sobremesa como el objeto portátil apuntan a null
- Al final sólo queda un objeto de tipo Ordenador con precio 1100
- Al final sólo queda un objeto de tipo Ordenador con precio 900
- Al final hay dos objetos de tipo Ordenador, uno con precio 900 y otro con precio 1100

11. Implemente un programa que muestre por pantalla el resultado de la suma y el producto de dos números complejos. Además, el programa deberá comprobar si dos números complejos dados son iguales y también deberá multiplicar un número complejo por un entero. Para ello:

- Dibuje el diagrama UML de la/s clase/s implicada/s, indicando todos sus miembros.
- Implemente el código de la/s clase/s en Java.
- Implemente un programa principal que muestre por pantalla el número complejo (en la forma $x+yi$) resultante de la suma y el producto de dos números complejos dados, así como el resultante de la multiplicación de un número complejo por un entero. Además, el programa deberá indicar si los dos números complejos introducidos son iguales.

Nota: Sean a y b dos números reales*, el par (a, b) se llama número complejo, si la igualdad, la adición y la multiplicación de pares se definen del modo siguiente:

a) *Igualdad*: $(a,b) = (c,d) \Leftrightarrow a = c \wedge b = d$

b) *Suma*: $(a,b) + (c,d) = (a+c, b+d)$

c) *Producto*: $(a,b) \cdot (c,d) = (ac - bd, ad + bc)$

El número complejo (a, b) puede expresarse en la forma $a+bi$. También se puede definir la operación multiplicación de un número complejo por un entero de la siguiente forma:

Multiplicación por un entero: $\alpha(a,b) = (\alpha a, \alpha b)$

* Por simplicidad considere a y b como enteros.

12. ¿Qué imprime el siguiente programa?

```
class Principal {
    public static void main(String[] args) {
        Autor autor =new Autor ("Paul", "Auster");
        Libro libro= new Libro ("La trilogía de Nueva York", autor);
        System.out.println (libro.getTitulo()+" de " +
            (libro.getAutor()).toString());
    }
}
class Libro {
    private String titulo;
    private Autor autor;
    public Libro(String titulo, Autor autor) {
        this.titulo = titulo;
        this.autor = autor;
    }
    public Autor getAutor() {
        return autor;
    }
    public String getTitulo() {
        return titulo;
    }
}
class Autor {
    private String nombre;
    private String apellido;
    public Autor (String nombre, String apellido) {
        this.nombre = nombre;
        this.apellido=apellido;
    }
    public String toString () {
        return nombre+ " " +apellido;
    }
}
```

13. ¿Qué imprime el siguiente programa?

```
class Alumno {
    String nombre = "Alumno 1";
    char grupo = 'A';
    char devolvergrupo() {
        return grupo;
    }
    void cambiargrupo(char nuevogrupo) {
        grupo = nuevogrupo;
    }
    void imprimir() {
        System.out.println("Nombre: " + nombre);
        System.out.println("Grupo: " + devolvergrupo());
    }
}
class Principal {
    public static void main (String[] args) {
        Alumno a1 = new Alumno();
        System.out.println("Objeto a1:\n-----\n");
        a1.imprimir();
        System.out.println("\nLlamada al metodo prueba1(a1).\n");
        prueba1(a1);
        a1.imprimir();
        System.out.println(" ");
        Alumno a2 = new Alumno();
        System.out.println("\nObjeto a2:\n-----\n");
        a2.imprimir();
        System.out.println("\nLlamada al metodo prueba2(a2).\n");
        prueba2(a2);
        System.out.println("\nEn main: Fuera ya del metodo prueba2. \n");
        a2.imprimir();
    }
    static void prueba1(Alumno b) {
        b.cambiargrupo('B');
    }
    static void prueba2(Alumno b) {
        b = new Alumno();
        b.cambiargrupo('B');
        System.out.println("\nDentro del metodo prueba2. \n");
        b.imprimir();
    }
}
```

14. En una tienda cada factura se representa por su número y una serie de productos. Cada producto se caracteriza por su código alfanumérico y su precio. Se pide:

- Dibujar el diagrama de clases indicando las relaciones entre las mismas y los miembros que hay en cada una. Debe incluir métodos para la consulta del número de factura y el/los productos de la factura y métodos para actualización de los datos del/los productos.
- Implemente en Java dichas clases
- Implemente un programa principal donde se imprima una factura con los siguientes productos: (código: C1, precio: 24 euros), (código: H23, precio: 234 euros) y (código: M30, precio: 109 euros). A continuación debe cambiarse el segundo producto de la factura por el siguiente (código: K123, precio:247 euros) y volver a imprimir la factura

15. Escriba un programa que use una lista enlazada para gestionar una serie de libros identificados por su título. El programa debe ser capaz de ir añadiendo libros a la serie mantenida y de poder imprimir los títulos de toda la serie cuando se le indique. Como ejemplo use los siguientes libros: "Niebla", "La Metamorfosis" y "El callejón de los milagros"

16. ¿ Qué imprime el siguiente programa?

```
class Colores {
    public static void main(String [] args) {
        Lista colores =new Lista();
        colores.incluir("rojo");
        colores.incluir("azul");
        colores.incluir("verde");
        colores.imprimir();
    }
} // Fin clase principal

class Nodo {
    private String color;
    private Nodo siguiente;
    public Nodo(String color) {
        this.color=color;
    }
    public void poner(Nodo siguiente_nodo) {
        siguiente=siguiente_nodo;
    }
    public Nodo coger() {
        return siguiente;
    }
    public String color() {
        return color;
    }
} // Fin clase Nodo

class Lista {
    private Nodo primero;
    public Lista() {
        primero=null;
    }
    public void imprimir() {
        for( Nodo aux=primero; aux.coger()!= null;aux = aux.coger()) {
            System.out.print(aux.color()+ " ");
        }
    }
    public void incluir(String cadena) {
        Nodo elemento = new Nodo(cadena);
        if (primero==null) {
            primero=elemento;
        }
        else {
            Nodo aux;
            for (aux=primero;aux.coger()!= null;aux=aux.coger());
            aux.poner(elemento);
        }
    }
} // Fin clase Lista
```

a) rojo azul verde

b) null

c) rojo azul

d) azul verde