

Gregorio Vidoy Fajardo

Ejercicio 3 P1

Tabla de contenidos:

DESCRIPCIÓN DEL PROBLEMA:	2
RELEASE NOTE:	3
SOLUCIÓN AL PROBLEMA:	4
EXPLICACIÓN DEL PROGRAMA:	4
UML:	5

DESCRIPCIÓN DEL PROBLEMA:

Utilizando el patrón que se considere más adecuado para recorrer una estructura de objetos, desarrollar un programa para generar presupuestos de configuración de un computador simple, formado por los siguientes elementos: Disco, Tarjeta, Bus. El programa mostrará el precio de cada posible configuración de Equipo:

```
public abstract class Equipo{

    private String nombre;

    public Equipo(String nombre){

        this.nombre= nombre;

    }

    public String nombre(){

        return nombre;

    }

    public abstract double potencia();

    public abstract double precioNeto();

    public abstract double precioConDescuento();

    public abstract void aceptar(VisitanteEquipo ve);

}
```

Las clases Disco, Tarjeta, Bus extienden a la clase abstracta Equipo e implementan todos sus métodos.

La programación del método aceptar(VisitanteEquipo ve) en cada una de las clases anteriores consistirá en una llamada al método correspondiente de la clase abstracta VisitanteEquipo:

```
public abstract class VisitanteEquipo{

    public abstract void VisitarDisco(Disco d);

    public abstract void VisitarTarjeta(Tarjeta t);

}
```

```
public abstract void VisitarBus(Bus b);  
  
}
```

Las subclases de VisitanteEquipo definirán algoritmos concretos que se aplican sobre la estructura de objetos que se obtiene de instanciar las subclases de Equipo. Por ejemplo, la subclase visitante: VisitantePrecio puede servir para calcular el coste neto de todas las partes que conforman un determinado equipo (disco+tarjeta+bus), acumulando internamente el costo de cada parte después de visitarla.

Además utilizando el patrón Visitante podemos adaptar la tabla de precios, que incluye a todos los componentes de un equipo, a diferentes tipos de clientes (clientes “VIP”, con descuento especial, etc.), simplemente cambiando la clase VisitantePrecio.

El programa Cliente se encarga de generar aleatoriamente el tipo de cliente, es decir: cliente sin-descuento, VIP (10% descuento), mayorista(15% descuento) y obtener el coste total de una configuración de equipo utilizando para ello sólo un objeto VisitanteEquipo.

Programar más subclases del tipo Visitante para mostrar los nombres de las partes que componen un equipo y sus precios. Ahora el programa Cliente mostrará, además del coste total de un equipo, las marcas de sus componentes.

Utilizando creación de hebras en Java, lanzar 100 consultas de precios concurrentes para diferentes tipos de clientes de tal manera que si producen más de 25 peticiones por parte de un mismo tipo de clientes se les haría un 5% de descuento adicional en el precio final de los equipos.

Presentar los resultados del programa en forma de tabla, en filas tipo de cliente, indicar: “regular”, “VIP”, “mayorista”; en las columnas: el número de peticiones, precio unitario y descuento aplicado para cada encargo.

RELEASE NOTE:

- Compilado con Java 1.8
- Clase Main Cliente.

SOLUCIÓN AL PROBLEMA:

-Para la resolución del problema he empleado el patrón visitante, creo una serie de Equipos especializados de la clase Equipo, Bus, Disco, Equipo. En los cuales se implementa el método aceptar, que abre el paso a los visitantes.

-El Visitante se define como abstracto del cual especializan los clientes Vip, Mayorista y SinDescuento, cada uno implementan el método visitar, para acceder al Equipo y obtener el dato requerido acumulándolo.

-Cliente extiende thread para implementar en el método run() la ejecución simultanea de 100 consultas.

EXPLICACIÓN DEL PROGRAMA:

El programa crea tres objetos, uno de cada tipo y un arraylist de hebras, y aleatoriamente inserta 100 visitantes, posteriormente se lanzan las hebras obteniendo los precios con descuentos en función de las peticiones realizadas.

UML:

