04 - EJERCICIO DE PROGRAMACIÓN JAVA

Servicios.

Práctica con Clases Abstractas

Empresa de Mantenimiento

Se va a realizar una aplicación que controle los servicios que realiza una empresa de mantenimiento. Estos servicios son muy variados, pero todos ellos comparten una serie de características comunes.

Así pues, se pide que programe una clase abstracta llamada Servicio que especificará esas características comunes y que servirá para crear, a través de herencia, las distintas clases de servicios que ofrece la empresa.

Clase Abstracta Servicio

Un servicio siempre tendrá las siguientes propiedades:

Trabajador (String) – nombre y apellidos del trabajador que realiza el servicio.

FechaInicio (LocalDate) – fecha de inicio del servicio.

Cliente (String) – Es el nombre y apellidos del cliente (o nombre de la empresa cliente)

Debe haber un constructor que reciba las tres propiedades anteriores.

Debe haber métodos set y get para estas dos propiedades.

Además, un servicio siempre debe tener programados los siguientes métodos:

double costeMaterial();

Este método calculará el total gastado en material.

double costeManoObra();

Este método calculará el total gastado en mano de obra.

double costeTotal();

Este método calculará el coste total del servicio.

String detalleServicio();

Este método devolverá una cadena con información detallada de lo que ha costado el Servicio

Clase TrabajoPintura

Esta clase describirá un trabajo de pintura (pintar una casa, una habitación, etc…) Heredará de la clase Servicio y tendrá las siguientes características:

Propiedades:

(Además de las de la clase Servicio)

Superficie (double) – Es la superficie a pintar (metros cuadrados).

PrecioPintura (double) – Es el precio de un litro de pintura.

Constructor

Crear un constructor que reciba: el trabajador que hace el servicio, la fecha de inicio, el cliente, la superficie y el precio del litro de pintura.

Métodos get y set

Programe un método get y set para las propiedades de la clase.

Métodos sobrescritos

Se sobreescribirán los métodos abstractos, dándoles un significado según la siguiente especificación:

double costeMaterial();

Según nos indica el cliente, el coste de material se calcula con la siguiente fórmula:

Coste_material = (Superficie / 7.8) * PrecioPintura;

double costeManoObra();

Según nos indica el cliente, el coste de mano de obra se calcula con la siguiente fórmula:

Coste_mano_obra = (Superficie / 10) * 9.5;

double costeTotal();

Según nos indica el cliente, el coste total del servicio se calcula así:

Coste_total = coste_material + coste_mano_obra;

Hay que tener en cuenta que cuando la superficie a pintar es de menos de 50 metros cuadrados se añade un coste adicional del 15% sobre el anterior coste.

String detalleServicio();

Este método devolverá un resumen del servicio con la siguiente estructura:

```
TRABAJO DE PINTURA
Cliente: (cliente)
Fecha de inicio: (fecha)
---------------------------------------
Pintor: (pintor)
Coste Material..... (coste material)
Coste Mano Obra.... (coste mano de obra)
Coste Adicional.... (coste adicional)
TOTAL: ............ (total coste servicio)
---------------------------------------
```

Clase RevisionAlarma

Uno de los servicios que realiza la empresa es la revisión de las alarmas contraincendios. Para definir este tipo de trabajo programe la clase RevisionAlarma, heredándola de Servicio, con las siguientes características:

Propiedades:

Solo tendrá una: el número de alarmas a revisar (int)

Constructor:

Un constructor que reciba: la fecha de la revisión, el nombre del cliente y el número de alarmas. Este constructor inicializará el nombre del trabajador a “Revisor Especialista Contraincendios”

Métodos get y set

Un get y set para el número de alarmas.

Métodos sobrescritos

double costeMaterial();

El coste de material en este tipo de trabajo es 0.

double costeManoObra();

El coste de mano de obra dependerá del número de alarmas a revisar y se calculará de la siguiente forma:

Coste_mano_obra = (número_alarmas / 3) * 40;

double costeTotal();

El coste total es igual al coste de mano de obra.

String detalleServicio();

Este método devolverá un resumen del servicio con la siguiente estructura:

```
REVISIÓN PERIÓDICA ALARMAS CONTRAINCENDIOS
Cliente: (cliente)
Fecha revisión: (fecha)
----------------------------------------
TOTAL: ..... (total coste servicio)
----------------------------------------
```

Programa de Prueba

Realice un programa de prueba en el que se creen varios trabajos de pintura y revisiones de alarmas.

Pruebe el concepto de polimorfismo almacenando todos estos trabajos dentro de un ArrayList de Servicio. Pruebe a calcular la suma de los costes de todos los trabajos. Calcule también lo que se tiene que pagar en total en sueldos por esos trabajos realizados.

Muestre en pantalla el resumen detallado de cada uno de esos trabajos.