IES Suarez de Figueroa, curso 2024-25

Examen 2 2ª Evaluación DAW1(Programación) 11/03/2025

Construir un programa que simula la gestión de una empresa de mantenimiento, que realiza servicios muy variados, pero todos 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 Empleado

Con los atributos:

Int id

String nombre

Constructor

Getters, Setters y toString

Clase Servicio

Un servicio siempre tendrá las siguientes propiedades:

idServicio(int)- entero que identifica un servicio

Empleado empleado– objeto de la clase Empleado

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 implementar los siguientes métodos, que aquí no tienen código:

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)

metrosCuadrados (double) – Es la superficie a pintar .

precioMetroCuadrado (double) – Es el precio de pintura por metro cuadrado.

Constructor

Crear un constructor que reciba: el empleado 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:

Método toString

Este método debe implementarse de tal forma que guarde el formato del archivo csv para poder mas tarde guárdalo.

double costeMaterial();

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

Coste_material = (metroCuadrado / 7.8) * precioMetrosCuadrado;

double costeManoObra();

Según nos indica el cliente, el coste de mano de obra viene recogida en el convenio, que viene implementada una clase estática llamada Convenio con un método llamado precioHora:

Crear una clase auxiliar estática, que tenga un método que devuelva la mano de obra según el servicio. Es decir, para TrabajoPintura

Coste_mano_obra = lo que diga el convenio

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.

Método toString

Este método debe implementarse de tal forma que guarde el formato del archivo csv para poder mas tarde guárdalo.

Métodos sobrescritos

double costeMaterial();

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

double costeManoObra();

Igual que la anterior pero esta es otra clase.

calculará de la siguiente forma:

Coste_mano_obra = número_alarmas * lo que marca el convenio;

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

Se entrega un fichero csv(servicios.csv) para cargar servicios de ejemplo, desde la clase GestionServicios. También se entrega la clase Main totalmente operativa y otras clases auxiliares.

(3)Construir la jerarquía de clases expuesta, teniendo en cuenta que los objetos Servicios se ordenaran por defecto o también llamado orden natural por el atributo fechaInicio, y que también debemos implementar los métodos correspondientes para saber si dos objetos Servicios son iguales atendiendo a que sean iguales si sus idServicios y su cliente son iguales

En la clase GestionServicios implementar los métodos para justificar el menú siguiente:

```
(0.5)"1 Nuevo Servicio\n"
(1)"2 Mostrar todos los empleados\n" +
(0.5)"3 Buscar Servicios de un empleado\n" +
(0.5)"4 Buscar Servicios de un cliente\n" +
(0.75)"5 Buscar Servicios de un tipo\n" +
(0.75)"6 Buscar si un Servicio está repetido\n" +
(0.75)"7 Eliminar Servicio por Id\n" +
(1)"8 Mostrar lista ordenada por fecha pero inversa\n" (orden
natural)+
(1.25)"9 Mostrar lista ordenada por cliente\n" +
"10 Guardar\n" (ya implementado en la clase+
"11 Salir\n");
```