Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE PROGRAMACIÓN II

24-MARZO-2021

Nombre: _____

DNI: _____ *Aula:* _____ *Fila:* _____ *Columna:* _____

1. Diseño

Debido al éxito en la campaña de vacunación, desde Deusto hemos decidido abrir una agencia de viajes que gestione excursiones y guías para atender a la avalancha de turistas que se prevé recibir en las próximas fechas. De cada sesión de guía necesitamos guardar el nombre de la persona que hace de guía (String), su precio por persona (double), el número de personas que irán a esa sesión (int) y si está pagada o todavía no (boolean), por ejemplo: `Guia("Pablo Garaizar", 3.5, 10, false)`. De cada excursión queremos almacenar el nombre de la excursión, la lista de trayectos que se realizan en la excursión y si está pagada o todavía no. Cada trayecto almacena la latitudOrigen (de -90.0 a 90.0), longitudOrigen (de -180.0 a 180.0), latitudDestino, longitudDestino, distancia (en km) y el transporte utilizado.

Para cada transporte queremos guardar el número de personas que admite y un código único generado automáticamente a partir de un contador común. Además, todos los transportes tendrán un método `getPrecioPorKm`() que se calculará de manera diferente para cada tipo de transporte. En el caso de los barcos, necesitaremos almacenar su calado (double) y si son diesel o no (boolean) y `getPrecioPorKm` se calculará multiplicando su calado por 1.5 si no es diesel o multiplicando su calado por 0.8 si es diesel. En el caso de las avionetas, necesitaremos almacenar su peso (double) y su coeficiente de planeo (double) y y `getPrecioPorKm` se calculará dividiendo su peso entre su coeficiente de planeo.

Tanto las excursiones como las guías serán cobrables, lo que implica que implementen dos métodos: 1) `calcularCoste`(), que devuelve un double con el coste total y 2) `cobrar`() que devuelve un boolean simulando que se ha producido el cobro. El coste de cada guía se calcula multiplicando el precio por persona por el número de personas, mientras que el coste de cada excursión se calcula sumando el coste de todos sus trayectos (distancia x `getPrecioPorKm`() del transporte usado). El método `cobrar` pondrá el atributo pagada de las guías y las excursiones a true y devolverá siempre true.

Además, en la clase Excursion necesitamos que haya un método `getExcursionMasCara`() que reciba un ArrayList de Excursiones y devuelva la Excursion de mayor coste del ArrayList recibido.

Dibuja un diagrama de clases para contemplar este caso.

Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE
PROGRAMACIÓN II

24-MARZO-2021

2. Implementación

1. Programa la jerarquía de clases diseñada en el apartado 1 en Java. Cada clase deberá tener **constructores** con argumentos y sin argumentos, **getters y setters** y **toString**.

2. Programa una clase DeustoTravel con un método main en el que crees al menos **3 guías y 3 excursiones** que tengan al menos **3 trayectos cada una** en diferentes medios de **transporte**.

3. Muestra desde el main de DeustoTravel cuál es la excursión más cara usando el método **`getExcursionMasCara`**() de la clase Excursion.

Extra: crea en el main de DeustoTravel una VentanaGrafica de 800x600 píxeles de resolución y muestra cada excursión dibujando una flecha por cada trayecto, desde el origen al destino (traduce la latitud a la altura disponible, donde -90.0 será 0 y 90.0 será la altura de la ventana, y lo mismo para la longitud: -180.0 será 0 y 180.0 será la anchura de la ventana). Sería deseable que por cada excursión generaras un color aleatorio y usaras ese color para que todas las flechas de esa excursión tengan ese mismo color.

Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE
PROGRAMACIÓN II

24-MARZO-2021