**Evaluación Parcial N°2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sigla** | **Nombre Asignatura** | **Tiempo Asignado** | **% Ponderación** |
| DSY1102 | Desarrollo Orientado a Objetos | **120 minutos** | **40%** |

1. **Situación evaluativa**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | **Ejecución práctica** |  |  | **Entrega de encargo** |  |  | **Presentación** |

1. **Agente evaluativo**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | **Heteroevaluación** |  |  | **Coevaluación** |  |  | **Autoevaluación** |

1. **Tabla de Especificaciones**

|  |  |  |
| --- | --- | --- |
| **Resultado de Aprendizaje** | **Indicador de Logro (IL)** | **Ponderación Indicador Logro** |
| RA2 Aplica conceptos avanzados del paradigma orientado a objetos en componentes de software, para dar respuesta a requerimientos de clientes. | IL 2.1 Aplica las sentencias de ciclos, para ser representadas en la solución según el requerimiento del usuario en un caso de negocios. | 15% |
| IL 2.2 Aplica conceptos de encapsulación avanzada permitiendo cambios internos sin afectar otros componentes del sistema del software. | 15% |
| IL 2.3 Configura una colección para almacenar información temporal en el programa según lo solicitado por el usuario. | 30% |
| IL 2.4 Aplica los conceptos de herencia en la programación orientada a objetos que permitan dar solución a un problema planteado un caso de negocios. | 20% |
| IL 2.5 Utiliza abstracción y polimorfismo de manera efectiva, para permitir la creación de interfaces flexibles y genéricas que puedan adaptarse a diferentes situaciones. | 20% |
| **Total** | | **100%** |

1. **Instrucciones generales para el/la estudiante**

|  |
| --- |
| Esta es una evaluación que corresponde a una ejecución práctica y tiene un 40% de ponderación sobre la nota final de la asignatura. El tiempo para desarrollar esta evaluación es de **120 minutos** y se realiza de manera **individual** en laboratorio PC avanzado.  Esta evaluación consiste en la resolución de un caso donde el estudiante debe aplicar colecciones y herencia. Por otro lado, realizar métodos abstractos, polimorfismos, sobrecarga e implementación de interface, entre otros.  Para dejar registro deberá realizar los siguientes puntos:   * Programar en el lenguaje JAVA utilizando el IDE NetBeans el diagrama de clases generado. * Deberá dejar registro del trabajo realizado por medio de la plataforma de Blackboad, para ello debe comprimir la carpeta de su proyecto y subirla a la actividad correspondiente en .ZIP |

1. **Evaluación**

**Excursiones de Turismo**

**Diagrama

El contenido generado por IA puede ser incorrecto.Se tiene la siguiente jerarquía de clases:**

Se necesita modelar un sistema para gestionar las diferentes excursiones que ofrece la agencia de turismo “Bucólico S.A.” La información que se conoce es la siguiente:

* **Aventuras**: deporte (rafting, escalada, trekking), equipamiento, código, nombre, duración (en horas), preciobase, dificultad (baja, media, alta).
* **Culturales**: código, nombre, duración (en horas), preciobase, dificultad (baja, media, alta), destinohistórico y idioma del guía.
* La interfaz **Promoción** contiene el atributo DTO\_TEM (15%) y el método aplicarDescuento. Este método aplicará el DTO\_TEM solo si la duración de la excursión es mayor a 5 horas y el nivel de dificultad es **alta**. En cualquier otro caso, el descuento será cero.

Además, se necesitan los siguientes métodos:

* **disminuirBase**: disminuirá el precio base en un 10% si el nivel de dificultad es **baja**.
* **calcularCostoAdicional**: aumentará el precio base en un X% si el deporte principal es **rafting** o si el idioma del guía es **inglés**. Y devolverá la cantidad aumentada.

Crear una clase manejadora, que se llamará Catalogo, deberá contener una colección de Excursion. Los métodos que esta clase debe tener son:

* agregarExcursion(Excursion): agrega una excursión al catálogo. Debe validar que no esté duplicada por su código.
* buscarExcursion(String): busca y devuelve una excursión por su código.
* aplicarAjusteATodas(): recorre el catálogo y aplica el método disminuirBase a todas las excursiones que cumplan con la condición.
* calcularDescuentoTotal(): recorre la colección y calcula el total de dinero ahorrado por concepto de DTO\_TEM.
* eliminarExcursion():por código, devolviendo un mensaje adecuado.

Construir las clases necesarias para resolver el problema planteado, no olvidar constructores, accesadores (get), mutadores (set). En el método principal (main), Implementar el siguiente menú de opciones:

Bucólico S.A:

1. **Ingresar Excursión**: permite agregar una Excursión de Aventura o Cultural.
2. **Mostrar Información**: mostrar toda la información de una excursión.
3. **Aplicar Ajuste de Precio**: ejecuta el método aplicarAjusteATodas.
4. **Eliminar excursión.**
5. **Salir**.

**Notas para el estudiante:**

* Considere el uso de clase y métodos **abstractos** donde sea necesario para modelar la jerarquía de manera correcta.