C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

SISTEMAS DE GESTIÓN EMPRESARIAL

TRABAJO DE EVALUACIÓN 2º TRIMESTRE

Ana Belén Regal Teijeira
anaregalteijeira@liceolapaz.net

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

Índice

Ana Belén Regal Teijeira
anaregalteijeira@liceolapaz.net

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

Ejercicio 1: Sistema de Gestión de Temperaturas por Ciudad

(Uso de una Única Clase y Almacenamiento en Listas, Tuplas y Diccionarios)

Objetivo: Crear un sistema para registrar temperaturas en diferentes ciudades y realizar cálculos estadísticos sobre las temperaturas de cada ciudad, tales como media, máxima y mínima. El sistema debe almacenar los datos en una única clase sin herencia, utilizando listas, tuplas y diccionarios.

Descripción:

El sistema debe gestionar las temperaturas registradas en varias ciudades, y realizar cálculos estadísticos (media, máxima, mínima) sobre los datos de temperatura. Cada ciudad tendrá un conjunto de temperaturas para cada mes del año.

Los tipos de datos que se deben utilizar son:

- **Listas**: Para almacenar las temperaturas registradas de cada ciudad.
- **Tuplas**: Para almacenar los datos de cada mes (como un conjunto de valores constantes para cada mes de la ciudad).
- **Diccionarios**: Para almacenar las ciudades como claves y sus temperaturas (listas de tuplas) como valores.

Acciones a realizar que se incluirán en un menú:

1. **Agregar una ciudad y sus temperaturas**: Permitir al usuario registrar una ciudad junto con las temperaturas de cada mes del año.
2. **Mostrar las temperaturas de una ciudad**: Mostrar las temperaturas registradas para una ciudad específica.
3. **Calcular la media de temperaturas de una ciudad**: Calcular la media de las temperaturas de una ciudad a partir de los datos registrados.
4. **Obtener la temperatura máxima y mínima de una ciudad**: Calcular la temperatura máxima y mínima de las temperaturas de una ciudad.
5. **Eliminar ciudad**: Permitir al usuario eliminar una ciudad y sus temperaturas del sistema.

Clase `Ciudad`:

Esta clase almacenará toda la información relacionada con las ciudades y sus temperaturas. A continuación, se detallan sus atributos y métodos.

Atributos de la clase:

- `nombre`: Nombre de la ciudad.

Ana Belén Regal Teijeira anaregalteijeira@liceolapaz.net

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

- `temperaturas`: Diccionario donde las claves son los meses (como números de 1 a 12) y los valores son las temperaturas registradas para ese mes. La lista de temperaturas debe ser una lista de tuplas (mes, temperatura).

Métodos de la clase:

1. `agregar_temperaturas(self, temperaturas):`
    - Método para agregar las temperaturas a la ciudad. Recibe una lista de temperaturas correspondientes a cada mes del año (12 meses).
2. `mostrar_temperaturas(self):`
    - Método para mostrar las temperaturas registradas para la ciudad, mostrando el mes y la temperatura.
3. `calcular_media(self):`
    - Método para calcular y retornar la media de las temperaturas registradas en la ciudad.
4. `calcular_maxima_minima(self):`
    - Método para calcular y retornar las temperaturas máxima y mínima registradas en la ciudad.
5. `eliminar_ciudad(self):`
    - Método para eliminar la ciudad y sus datos del sistema. Este método no solo eliminará los datos de la ciudad, sino también las temperaturas asociadas.

Acciones a realizar en el menú:

1. **Menú interactivo**: El sistema debe mostrar un menú interactivo que permita al usuario elegir entre las siguientes opciones:
    - Registrar una ciudad y sus temperaturas.
    - Mostrar las temperaturas de una ciudad.
    - Calcular la media de las temperaturas de una ciudad.
    - Calcular la temperatura máxima y mínima de una ciudad.
    - Eliminar una ciudad.
    - Salir.
2. **Implementación de operaciones CRUD**:
    - **Registrar ciudad y temperaturas**: Solicitar al usuario el nombre de la ciudad y las temperaturas de cada uno de los 12 meses del año. Los datos de las temperaturas deben almacenarse como una lista de tuplas, donde cada tupla contiene el mes y la temperatura correspondiente.

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

- **Mostrar temperaturas de una ciudad**: El sistema debe permitir al usuario ver las temperaturas registradas para una ciudad específica.
- **Calcular la media de temperaturas**: La función debe calcular y devolver la media de las temperaturas de la ciudad.
- **Obtener la temperatura máxima y mínima**: Calcular y devolver las temperaturas máxima y mínima de los datos registrados para la ciudad.
- **Eliminar ciudad**: El sistema debe permitir eliminar una ciudad del sistema, eliminando tanto la ciudad como sus datos de temperaturas.

Estructura del Diccionario de Almacenamiento:

El diccionario que almacena las ciudades debe tener la siguiente estructura:

```
{
"Madrid": [ (1, 15), (2, 16), (3, 18), ..., (12, 10) ], # Tupla (mes,
temperatura) "Barcelona": [ (1, 17), (2, 17), (3, 19), ..., (12, 11) ]
}
```

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

Ejercicio 2: Sistema de Gestión de Biblioteca

(Clases, Herencia, CRUD y Almacenamiento en Listas)

Objetivo: Crear un sistema de gestión de libros en una biblioteca que permita gestionar tanto libros físicos como electrónicos. El sistema debe usar clases y herencia, permitiendo realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los libros. Los datos de los libros deben almacenarse en una lista.

Descripción:

El sistema debe permitir la gestión de dos tipos de libros:

- **Libros Físicos**: Tienen un atributo adicional `ubicacion`, que indica su ubicación en el estante de la biblioteca.
- **Libros Electrónicos**: Tienen un atributo adicional `formato`, que indica el formato del libro (por ejemplo, PDF, EPUB, etc.).

El sistema permitirá realizar las siguientes acciones:

1. **Agregar un libro**: El sistema debe permitir agregar tanto libros físicos como electrónicos, solicitando los datos necesarios según el tipo de libro.
2. **Mostrar todos los libros**: El sistema debe mostrar todos los libros registrados con sus detalles (título, autor, año, y atributos específicos de cada tipo).
3. **Actualizar un libro**: El sistema debe permitir modificar los datos (título, autor, año) de un libro existente.
4. **Eliminar un libro**: El sistema debe permitir eliminar un libro del sistema.

Clases y Atributos:

- **Clase `Libro` (Clase Base):**
    - **Atributos**:
        - `titulo`: El título del libro.
        - `autor`: El autor del libro.
        - `año`: El año de publicación del libro.
    - **Funciones**:
        - `mostrar_info()`: Muestra la información básica del libro (título, autor y año).
- **Clase `LibroFisico` (Hereda de `Libro`):**

Ana Belén Regal Teijeira anaregalteijeira@liceolapaz.net

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

- **Atributos adicionales**:
  - `ubicacion`: Ubicación del libro en el estante de la biblioteca.
- **Funciones**:
  - `mostrar_info()`: Sobrescribe la función de la clase padre para incluir la ubicación del libro físico.

- **Clase `LibroElectronico` (Hereda de `Libro`):**
  - **Atributos adicionales**:
    - `formato`: Formato del libro electrónico (por ejemplo, PDF, EPUB, etc.).
  - **Funciones**:
    - `mostrar_info()`: Sobrescribe la función de la clase padre para incluir el formato del libro electrónico.

Acciones a realizar:

1. Crear un menú interactivo que permita al usuario elegir entre las opciones:
   - Agregar un libro.
   - Mostrar todos los libros.
   - Eliminar un libro.
   - Actualizar un libro.
   - Salir.
2. Implementar las funciones de agregar, mostrar, actualizar y eliminar libros:
   - **Agregar un libro**: Solicitar al usuario el tipo de libro (físico o electrónico), y luego los datos correspondientes.
   - **Mostrar libros**: Mostrar todos los libros almacenados en la lista.
   - **Eliminar libro**: Permitir al usuario seleccionar el libro a eliminar de la lista.
   - **Actualizar libro**: Permitir al usuario modificar los datos de un libro existente.

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

Ejercicio 3: Sistema de Gestión de Vehículos

(Clases, Herencia, CRUD y Almacenamiento en Diccionarios)

Objetivo: Crear un sistema de gestión de vehículos para una tienda de vehículos, donde puedas gestionar tanto coches como motos. El sistema debe almacenar los vehículos en un diccionario usando el ID como clave.

Descripción:

El sistema debe permitir gestionar dos tipos de vehículos:

- **Coches**: Tienen un atributo adicional `numero_de_puertas`, que indica el número de puertas del coche.
- **Motos**: Tienen un atributo adicional `tipo_de_motor`, que indica el tipo de motor de la moto (por ejemplo, 2T, 4T, eléctrico).

El sistema debe permitir realizar las siguientes acciones:

1. **Agregar un vehículo**: El sistema debe permitir agregar tanto coches como motos, solicitando los datos necesarios y los atributos específicos.
2. **Mostrar todos los vehículos**: El sistema debe mostrar todos los vehículos registrados con sus detalles (ID, marca, modelo, año, y atributos específicos de cada tipo).
3. **Actualizar un vehículo**: El sistema debe permitir modificar los datos de un vehículo existente (marca, modelo, año).
4. **Eliminar un vehículo**: El sistema debe permitir eliminar un vehículo del sistema.

Clases y Atributos:

- **Clase `Vehiculo` (Clase Base):**
    - **Atributos**:
        - `id_vehiculo`: ID único del vehículo.
        - `marca`: Marca del vehículo.
        - `modelo`: Modelo del vehículo.
        - `año`: Año de fabricación del vehículo.
    - **Funciones**:
        - `mostrar_info()`: Muestra la información básica del vehículo (ID, marca, modelo, año).
- **Clase `Coche` (Hereda de `Vehiculo`):**

Ana Belén Regal Teijeira anaregalteijeira@liceolapaz.net

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

    - **Atributos adicionales**:
        - `numero_de_puertas`: Número de puertas del coche.
    - **Funciones**:
        - `mostrar_info()`: Muestra los atributos básicos y el número de puertas.
- **Clase `Moto` (Hereda de `Vehiculo`):**
    - **Atributos adicionales**:
        - `tipo_de_motor`: Tipo de motor de la moto (por ejemplo, 2T, 4T, eléctrico).
    - **Funciones**:
        - `mostrar_info()`: Muestra los atributos básicos y el tipo de motor.

Acciones a realizar:

1. Crear un menú interactivo con las opciones:
    - Agregar un vehículo.
    - Mostrar todos los vehículos.
    - Eliminar un vehículo.
    - Actualizar un vehículo.
    - Salir.
2. Implementar las funciones de agregar, mostrar, actualizar y eliminar vehículos:
    - **Agregar un vehículo**: Solicitar el tipo de vehículo (coche o moto) y los datos correspondientes.
    - **Mostrar vehículos**: Mostrar todos los vehículos almacenados en el diccionario.
    - **Eliminar vehículo**: Permitir al usuario seleccionar el vehículo a eliminar del diccionario.
    - **Actualizar vehículo**: Permitir al usuario modificar los datos de un vehículo existente.

C/ Sebastián Martínez Risco, 12
15009 – A Coruña
Email: info@liceolapaz.com

Calificación:

Este trabajo representa un 20% de la nota de evaluación del 2º trimestre. La puntuación se repartirá del siguiente modo:

Reparto de puntuación:

33% → cada uno de los ejercicios.

El día de la entrega, en clase, se realizará la modificación planteada y que será llave para la corrección de este boletín. Si la modificación no se realiza o no se realiza de forma correcta el boletín no se corregirá y por lo tanto la nota será de un 0. Si la modificación planteada es correcta se defenderán los ejercicios y el boletín será corregido.

El trabajo debe ser un trabajo original. El plagio es una violación del código de honor, Tómate tu tiempo para escribir tus propias ideas y palabras, los trabajos plagiados serán excluidos de la corrección, su puntuación será de un 0.

Cómo entregar:

Documento comprimido en ".zip" con la siguiente nomenclatura <<B01-2T-nombre-completo.zip>> que incluirá los tres proyectos correspondientes a cada uno de los ejercicios enunciados.

Cada alumno/a ha de entregar la documentación correspondiente en la tarea correspondiente de la asignatura de "Sistemas de Xestión Empresarial" que se publicará en la plataforma de trabajo de clase.

Plazo de entrega:

El trabajo se expondrá en las sesiones de clase del lunes 24 de febrero de 10:10 a 12:20.
En la primera sesión se realizará la modificación y en la segunda la defensa y ejecución.
El trabajo completo será entregado en la plataforma antes de las 10:00 de la mañana de ese mismo día.

Modificación a realizar en clase:

...