**GestionAeropuertos** tiene como objetivo gestionar datos relacionados con aeropuertos, vuelos y compañías aéreas. La clase carga datos de ejemplo y proporciona métodos para mostrar los vuelos de salida y llegada de cada aeropuerto.

**Atributos**

* aeropuertos: Lista estática que almacena objetos de tipo Aeropuerto.
* vuelos: Lista estática que almacena objetos de tipo Vuelo.
* companias: Lista estática que almacena objetos de tipo Compania.

**Métodos Principales**

1. **Método main**
   * Este es el punto de entrada del programa. Llama a otros métodos para cargar datos y mostrar información sobre vuelos.
   * Llama a cargarDatos(), mostrarVuelosSalida() y mostrarVuelosLlegada().
2. **Método cargarDatos**
   * Carga datos de ejemplo en las listas aeropuertos, vuelos y companias.
   * Añade aeropuertos a la lista aeropuertos.
   * Crea objetos Vuelo y los añade a la lista vuelos, asociándolos con aeropuertos y compañías existentes o nuevos.
3. **Método buscarAeropuerto**
   * Busca un aeropuerto por su código en la lista aeropuertos.
   * Devuelve el aeropuerto si lo encuentra, de lo contrario, devuelve null.
4. **Método buscarOcrearCompania**
   * Busca una compañía por su nombre en la lista companias.
   * Si la compañía existe, la devuelve. Si no, crea una nueva compañía, la añade a la lista y luego la devuelve.
5. **Método mostrarVuelosSalida**
   * Muestra todos los vuelos de salida desde cada aeropuerto.
   * Recorre la lista aeropuertos y para cada aeropuerto, imprime los vuelos cuya origen coincide con ese aeropuerto.
6. **Método mostrarVuelosLlegada**
   * Muestra todos los vuelos de llegada a cada aeropuerto.
   * Recorre la lista aeropuertos y para cada aeropuerto, imprime los vuelos cuya destino coincide con ese aeropuerto.

**Compania** :representa una compañía aérea con un solo atributo: su nombre.

**Atributos**

* nombre: Un atributo de tipo String que almacena el nombre de la compañía.

**Constructores**

* **Constructor Compania(String nombre)**
  + Este constructor recibe un parámetro nombre y lo asigna al atributo nombre de la instancia de Compania.
  + Permite crear un objeto Compania con un nombre específico.

**Métodos**

1. **Getter getNombre**
   * Devuelve el valor del atributo nombre.
   * Permite acceder al nombre de la compañía desde fuera de la clase.
2. **Setter setNombre**
   * Establece el valor del atributo nombre.
   * Permite cambiar el nombre de la compañía desde fuera de la clase.
3. **Método toString**
   * Sobrescribe el método toString de la clase Object.
   * Devuelve una representación en forma de cadena del objeto Compania, incluyendo el nombre.
   * Este método es útil para imprimir el objeto Compania de una manera legible.

**Vuelo:** representa un vuelo aéreo con varios atributos, incluyendo el tipo de vuelo, origen, destino, duración, compañía, tipo de avión, capacidad y horario.

**Atributos**

* tipo: Un atributo de tipo String que almacena el tipo de vuelo (por ejemplo, "R" para regular o "C" para chárter).
* origen: Un objeto de tipo Aeropuerto que representa el aeropuerto de origen del vuelo.
* destino: Un objeto de tipo Aeropuerto que representa el aeropuerto de destino del vuelo.
* duracion: Un atributo de tipo String que almacena la duración del vuelo.
* compania: Un objeto de tipo Compania que representa la compañía aérea que opera el vuelo.
* tipoAvion: Un atributo de tipo String que almacena el tipo de avión utilizado en el vuelo.
* capacidad: Un atributo de tipo int que representa la capacidad del avión en términos de número de pasajeros.
* horario: Un atributo de tipo String que almacena el horario del vuelo.

**Constructores**

* **Constructor Vuelo**
  + Este constructor recibe parámetros para inicializar todos los atributos de la clase.
  + Permite crear un objeto Vuelo con todos los detalles especificados.

**Métodos**

1. **Getters**
   * getTipo: Devuelve el tipo de vuelo.
   * getOrigen: Devuelve el aeropuerto de origen.
   * getDestino: Devuelve el aeropuerto de destino.
   * getDuracion: Devuelve la duración del vuelo.
   * getCompania: Devuelve la compañía aérea del vuelo.
   * getTipoAvion: Devuelve el tipo de avión del vuelo.
   * getCapacidad: Devuelve la capacidad del avión.
   * getHorario: Devuelve el horario del vuelo.
2. **Setters**
   * setTipo: Establece el tipo de vuelo.
   * setOrigen: Establece el aeropuerto de origen.
   * setDestino: Establece el aeropuerto de destino.
   * setDuracion: Establece la duración del vuelo.
   * setCompania: Establece la compañía aérea del vuelo.
   * setTipoAvion: Establece el tipo de avión del vuelo.
   * setCapacidad: Establece la capacidad del avión.
   * setHorario: Establece el horario del vuelo.
3. **Método toString**
   * Sobrescribe el método toString de la clase Object.
   * Devuelve una representación en forma de cadena del objeto Vuelo, incluyendo todos sus atributos.
   * Este método es útil para imprimir el objeto Vuelo de una manera legible.

**Aeropuerto**: representa un aeropuerto con varios atributos, incluyendo un código, nombre, población, país y GMT (Greenwich Mean Time, el desfase horario respecto al meridiano de Greenwich).

**Atributos**

* codigo: Un atributo de tipo String que almacena el código del aeropuerto.
* nombre: Un atributo de tipo String que almacena el nombre del aeropuerto.
* poblacion: Un atributo de tipo String que almacena la población donde se encuentra el aeropuerto.
* pais: Un atributo de tipo String que almacena el país donde se encuentra el aeropuerto.
* gmt: Un atributo de tipo int que almacena el desfase horario del aeropuerto respecto al GMT.

**Constructores**

* **Constructor Aeropuerto**
  + Este constructor recibe parámetros para inicializar todos los atributos de la clase.
  + Permite crear un objeto Aeropuerto con todos los detalles especificados.

**Métodos**

1. **Getters**
   * getCodigo: Devuelve el código del aeropuerto.
   * getNombre: Devuelve el nombre del aeropuerto.
   * getPoblacion: Devuelve la población donde se encuentra el aeropuerto.
   * getPais: Devuelve el país donde se encuentra el aeropuerto.
   * getGmt: Devuelve el desfase horario del aeropuerto respecto al GMT.
2. **Setters**
   * setCodigo: Establece el código del aeropuerto.
   * setNombre: Establece el nombre del aeropuerto.
   * setPoblacion: Establece la población donde se encuentra el aeropuerto.
   * setPais: Establece el país donde se encuentra el aeropuerto.
   * setGmt: Establece el desfase horario del aeropuerto respecto al GMT.
3. **Método toString**
   * Sobrescribe el método toString de la clase Object.
   * Devuelve una representación en forma de cadena del objeto Aeropuerto, incluyendo todos sus atributos.
   * Este método es útil para imprimir el objeto Aeropuerto de una manera legible.