Profesor: Dr. Oldemar Rodríguez Rojas
Paradigmas de Programación
Programación Orientada a Objetos en Lenguaje Python - Parte 1
Fecha de Entrega: Lunes 31 de octubre - 8am

Tarea Número 10

- Las tareas son estrictamente de carácter individual, tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso, es decir, el promedio de las notas obtenidas en la tareas será la nota final del curso.
- Todas las preguntas tienen el mismo puntaje.
- Incluir al menos 3 casos de prueba en las funciones programadas (cuando corresponda).
- **Nota:** Todas la programación de objetos tiene que ser realizada con **Estilo Pythonico**, programación estilo Java o C++ tendrá cero puntos.

Escriba en Python con los siguientes ejercicios con al menos una prueba de ejecución de cada clase. Todas las clases debe seguir el estilo correcto de python y poseer un método __str__.

1. Desarrolle en Python las clases Triángulo y Rectángulo, luego para cada una de ellas programe los métodos `calcular_area` y `calcular_perimetro`:
   - `Clase Triángulo`: Tiene como atributos el lado, la base y altura del triángulo.
   - `Clase Rectángulo`: Tiene como atributos la base y altura del rectángulo.
2. Desarrolle en pyhon la clase `Operación`, que tiene como atributos dos vectores numéricos U y V luego programe los métodos:
   - `Sumar`: Suma ambos vectores y devuelve el resultado.
   - `Restar`: Resta al primer vector el segundo y devuelve el resultado.
   - `Multiplicar`: Calcula el producto punto entre ambos vectores y devuelve el resultado. Es decir, si $U = (u_1, u_2, \ldots, u_n)$ y $V = (v_1, v_2, \ldots, v_n)$ entonces:

$$U \cdot V = u_1v_1 + u_2v_2 + \cdots + u_nv_n = \sum_{i=1}^{n} u_iv_i.$$

   Por ejemplo, si:

$$U = \left(1, \frac{1}{2}, 3\right), \quad V = (4, -4, 1),$$

   entonces:

$$U \cdot V = \left(1, \frac{1}{2}, 3\right) \cdot (4, -4, 1) = (1)(4) + \left(\frac{1}{2}\right)(-4) + (3)(1) = 4 - 2 + 3 = 5.$$

- `Correlacion`: Devuelve la correlación entre los dos vectores.
- `Covarianza`: Devuelve la covarianza entre los dos vectores.

3. Programe una clase en Python (estilo pythónico) denominada `Jugadores` que tiene como atributo un dataframe de `pandas` que permite almacenar una tabla de datos como la incluida en el archivo `Players1.csv`. Además, programe los siguientes métodos:

   - `actualizar_position`: Recibe el nombre del jugador y el nombre de la nueva posición, dicho método actualiza la posición del jugador y la muestra.
   - `resumen_jugador`: Recibe el nombre del jugador y retorna toda su información en formato de texto (`str`).
   - `resumen_columna`: Recibe el nombre de una columna y si es numérica retorna su promedio, en caso de ser categórica (`object`) retorna la moda.
   - `cantidad_equipos`: Retorna la cantidad de jugadores que existen en el dataframe por equipo, ejemplo Algeria: `10 jugadores`.
   - `agregar_jugador`: Recibe el nombre del jugador y los valores de cada columna del dataframe, añade el jugador a la tabla de datos y retorna un mensaje donde se indica que se agregó correctamente. En el siguiente link se muestran algunas maneras de agregar filas a un dataframe: agregar filas.
   - `eliminar_jugador`: Recibe el nombre del jugador, lo elimina de la tabla de datos y retorna un mensaje donde se indica que se eliminó correctamente. Debe validar que el jugador exista dentro de la tabla de datos.

   Verifique la correctitud de esta clase usando la tabla `Players1.csv` como atributo.

4. Desarrolle una clase denominada `Análisis` que tiene como atributo una matriz numérica tipo `numpy` y defina los siguientes métodos:

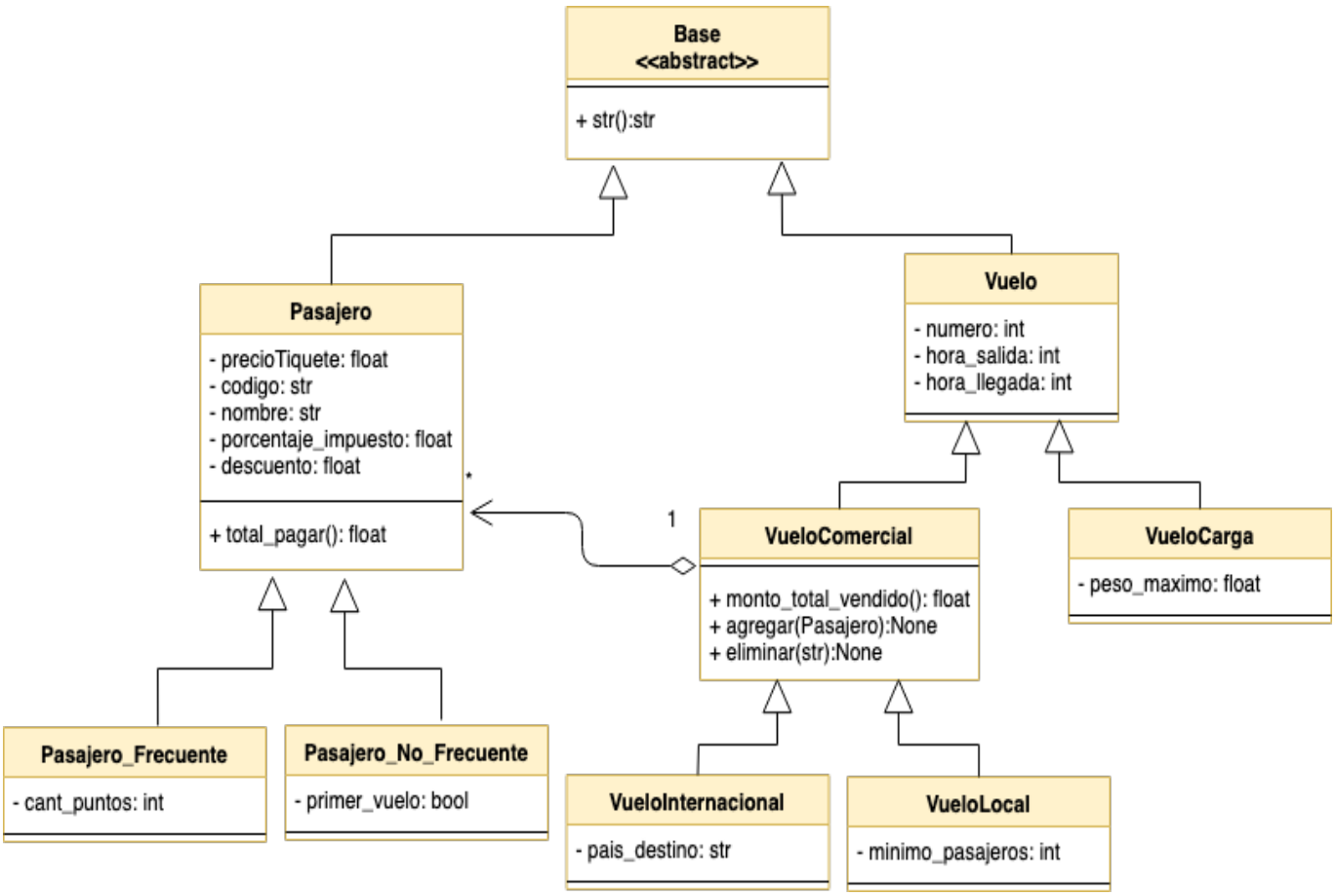
   - `as_data_frame`: retorna la matriz convertida en DataFrame de `pandas`.
   - `desviacion_estandar`: retorna un diccionario con la Desviación Estándar de cada columna.
   - `varianza`: retorna un diccionario con la Varianza de cada columna.
   - `moda`: retorna un diccionario con la Moda de cada columna.
   - `maximo`: retorna el máximo de toda la matriz.
   - `buscar`: recibe un número y busca el valor en la matriz, retorna los índices del primer valor encontrado o None en caso de no encontrar el valor.

5. Una línea aérea desea implementar un sistema para el control de sus vuelos, para esto se cuenta con la siguiente información:

   - Se supone que un Vuelo tiene los siguientes atributos: Número, Hora de Salida y Hora de Llegada.
   - Un Vuelo Comercial tiene además (respecto a un Vuelo) una lista de pasajeros Pasajeros.

- Un Vuelo Local (USA) tiene además (respecto a un Vuelo Comercial) un Número Mínimo de Pasajeros.
- Un Vuelo Internacional tiene además (respecto a un Vuelo Comercial) un País Destino.
- Un Vuelo de Carga tiene además (respecto a un Vuelo) un Peso Máximo de carga soportado.
- Un Pasajero tiene Código, Nombre, Precio Boleto, Porcentaje Impuesto y Total a Pagar = Precio Boleto + Porcentaje Impuesto * Precio Boleto. Los pasajeros son de dos tipos: los Pasajero Frecuente y los No Frecuentes, la diferencia es que a los pasajeros frecuentes se les aplica un 25 % de descuento y al No Frecuente se le aplica un 10 % de descuento en el Total a Pagar.
- En las clases vuelo local y vuelo internacional deben de poder eliminar y agregar pasajeros.

Programe una Jerarquía de Clases usando herencia que incluya al menos las clases: Vuelo,Vuelo Comercial, Vuelo Local, Vuelo Internacional, Vuelo de Carga, Pasajero, Pasajero no Frecuente y Pasajero Frecuente. Según el siguiente diagrama de clases:



6. Agregue a la clase `class mi_DF()` vista en clase los siguientes métodos:

   - Retorna la cantidad de valores de este DataFrame que son divisibles entre 5 (Pruebe este método leyendo un archivo de datos, esto en el Script de pruebas).

- Recibe dos números de columna y que retorna en un diccionario con el nombre de las variables correspondientes a las columnas, la covarianza y la correlación entre esas dos variables (Pruebe este método leyendo un archivo de datos, esto en el Script de pruebas).
- Reescriba la clase `class mi_DF()` pero heredando de la clase `pandas` en lugar de tener una relación componente-compuesto con `pandas`.

Entregables: Genere desde `Jupyter` un documento autoreproducible con la solución de la tarea y súbalo en el Aula Virtual, incluya casos de prueba en todos los ejercicios.