Última versión del documento disponible en:
https://docs.google.com/document/d/1izNQ8YD4Fk6UPzytYgyiK6gpHU3FMg0troVP_wHr4tE/edit?usp=sharing

Trabajo Práctico

Este trabajo tiene como objetivo afianzar la práctica de Programación Orientada a Objetos. En particular, de los mecanismos de herencia, polimorfismo, clases abstractas, interfaces, entrada/salida y patrones de diseño.
Fecha de entrega: 14 de Noviembre de 2023

1. Pautas de Trabajo

Tamaño de los grupos: de cuatro a seis integrantes.

2. Ejercicio: Héroes y Villanos

2.1. Planteo

Se desea modelar un juego compuesto por héroes y villanos. Cada personaje del juego posee un nombre real, un nombre de superhéroe o villano y un conjunto de características.
Las características son: velocidad, fuerza, resistencia y destreza.
Cada una de estas posee un nivel asociado, por ejemplo, velocidad: 500, fuerza: 700, etc.

El mecanismo básico del juego se basa en enfrentar un personaje con otro y decidir cuál de ellos es el ganador. Para decidir quién es el ganador se utiliza el valor de una de las características. En caso de empate, se decide por el valor de otra característica dada (en el orden establecido, y volviendo a comenzar si fuera necesario).
Por ejemplo: Si dos contendientes decidieran competir por Fuerza, y empatan, definen por Resistencia. Si hubiera otro empate, definen por Destreza. Ante otro empate, definen por Velocidad. Si resulta en empate, será empate finalmente.
El orden es:

1. Velocidad
2. Fuerza
3. Resistencia
4. Destreza

A su vez, el juego debe proveer un mecanismo de agrupamiento de los personajes en ligas para realizar desafíos o enfrentamientos entre grupos de personajes. Para esto, el valor

grupal de cada característica se determina como el promedio de los valores de esa característica entre todos los personajes.

Las ligas se almacenan en un archivo de texto, y se cargan en memoria antes de un enfrentamiento.
Los archivos tienen el siguiente formato:

```
personajes.in
Héroe/Villano, NombreReal, NombrePersonaje, Velocidad, Fuerza, Resistencia, Destreza
Héroe, Edward Blake, The Comedian, 100, 200, 150, 50
Villano, Adrian Veidt, Ozymandias, 120, 180, 200, 200
...
```

Liga “ligas.in” (archivo que contiene los personajes que la conforman. Puede contener otra Liga y Personajes sueltos)

```
ligas.in
Watchmen, The Comedian, Rorschach
MinuteMen, Captain Metropoli, Dollar Bill, Hollis Mason
MightPower, Watchmen, Sally Silk
...
```

Las Ligas son homogéneas, sólo de Héroes o sólo Villanos. Cada personaje sólamente puede pertenecer a una liga (o a ninguna).

Hacer un adecuado tratamiento de excepciones al construir Personajes y Ligas.

Es posible que también un solo personaje se enfrente a un grupo, o que una liga pertenezca a una liga más grande.
Se debe proveer servicios que permitan obtener:
• Todos los personajes/ligas que existen en el juego que venzan a un personaje dado para una cierta característica.
• Decidir quién es el vencedor de una disputa, acorde a una característica.

```
public boolean esGanador(Competidor competidor, Caracteristica c){}
```

• Poder armar ligas de súper héroes o súper villanos.

```
public Liga(){
     public boolean agregarCompetidor(Competidor competidor){...}
}
```

• Obtener listados de personajes ordenados (ascendente o descendentemente) por múltiples características. En caso de igualdad de dos personajes para una característica dada se debe permitir ordenarlos por las siguientes características seleccionadas.
Por ejemplo:
Por fuerza y luego por velocidad o por velocidad y luego por destreza.

2.2 Uso del Sistema

Todo se realizará por medio de la consola, no se esperan interfaces gráficas.

Solo se enfrentan Héroes contra Villanos (o ligas de Héroes contra Ligas de Villanos)
Las Ligas son homogéneas, sólo de Héroes o sólo Villanos.

El menú principal deberá permitir:

- Administración de Personajes
    - Carga desde archivo
    - Creación
    - Listado
    - Guardar en archivo todos los personajes
- Administración de Ligas
    - Carga desde archivo
    - Creación
    - Listado
    - Guardar en archivo todas las ligas
- Realización de combates
    - Personaje contra Liga (definiendo característica)
    - Liga contra Liga (definiendo característica)
- Reportes
    - Todos los personajes o ligas que venzan a un personaje dado para cierta característica
    - Listado ordenado de personajes por múltiples características

2.3 Temas a investigar

- Entrada/salida (buffer y scanner)
- System.in
- Comparable/comparator
- Patrones de Diseño

3. Qué entregar

Para la entrega, se deberá disponer de los siguientes entregables:

- **Diagrama de clases** actualizado (.jpg)
- **Pruebas unitarias** del código. Dichas pruebas deberán validar *significativamente* los aspectos críticos del programa.
- El **código:**
    - debidamente comentado (lease: siempre y cuando se evidencie necesario y no sea redundante ni trivial),
    - debidamente formateado, y
    - sin errores de compilación.
- Un **main** que evidencie el funcionamiento del programa.
- Los suficientes **archivos de entrada** de prueba *significativamente distintos*, para poder ejecutar distintas versiones del problema. Deberán totalizar al menos treinta personajes.
- Un **breve informe** (.pdf) que explique:
    - **integrantes**
    - **decisiones de diseño** (cuáles aspectos del problema consideraron relevantes para el diseño y cuáles no, qué opciones de implementación evaluaron, cuál fue la alternativa elegida en cada caso, y por qué), **sin entrar en detalles de implementación**
    - **descripción de cada archivo** .java comprendido en la solución del problema

- **organización y distribución del trabajo**
- **conclusiones**

Opcionalmente, también puede agregarse cualquier otro aspecto que consideren relevante (Ej: explicar el funcionamiento general de la solución para un ejemplo concreto).