Informe de Laboratorio 07

Tema: Arreglos Estandar y ArrayList

Nota

Estudiante	Escuela	Asignatura
Jhonatan David Arias Quispe jariasq@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de Programacion 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
07	Arreglos Estandar y ArrayList	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 18 Octubre 2023	Al 23 Octubre 2023

1. Actividades

- Cree un Proyecto llamado Laboratorio7
- Usted deberá crear las dos clases Soldado.java y VideoJuego4.java. Puede reutilizar lo desarrollado en Laboratorios anteriores.
- Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero).
- El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Para el tablero utilizar la estructura de datos más adecuada.
- Tendrá 2 Ejércitos (utilizar la estructura de datos más adecuada). Inicializar el tablero con n soldados aleatorios entre 1 y 10 para cada Ejército. Cada soldado tendrá un nombre autogenerado: Soldado0X1, Soldado1X1, etc., un valor de puntos de vida autogenerado aleatoriamente [1..5], la fila y columna también autogenerados aleatoriamente (no puede haber 2 soldados en el mismo cuadrado). Se debe mostrar el tablero con todos los soldados creados y sus puntos de vida (usar caracteres como y otros y distinguir los de un ejército de los del otro ejército). Además de los datos del Soldado con mayor vida de cada ejército, el promedio de puntos de vida de todos los soldados creados por ejército, los datos de todos los soldados por ejército en el orden que fueron creados y un ranking de poder de todos los soldados creados por ejército (del que tiene más nivel de vida al que tiene menos) usando 2 diferentes algoritmos de ordenamiento. Finalmente, que muestre qué ejército ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla). Hacer el programa iterativo.

2. Equipos, materiales y temas utilizados

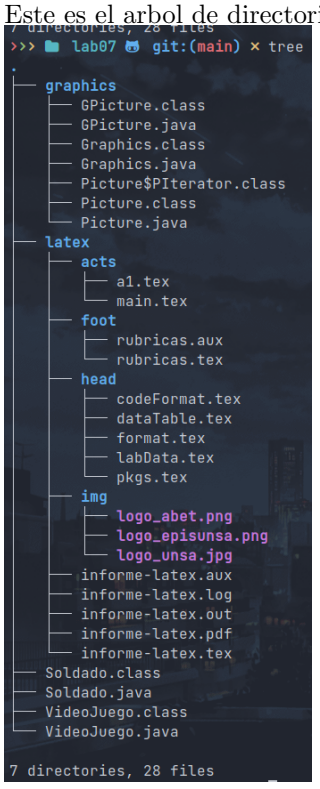
- Sistema Operativo ArchCraft GNU Linux 64 bits Kernell
- NeoVim
- OpenJDK 64-Bit 20.0.1
- Git 2.42.0
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Creacion de programas con CLI
- Bilioteca Graphics (origen propio)

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/JhonatanDczel/fp2-23b.git`
- URL para el laboratorio 07 en el Repositorio GitHub.
- `https://github.com/JhonatanDczel/fp2-23b/tree/main/fase02/lab07`

4. Orden de archivos

Este es el arbol de directorios final



5. Actividades iniciales: Crea un proyecto llamado Laboratorio 7

Se creo el directorio lab07, y se copio a el los archivos de los anteriores laboratorios para reciclar partes de codigo

- Se importaron las clases Soldado y Videojuego
- El soldado tiene las especificaciones requericas como atributos
  - Nombre
  - Puntos de vida
  - Fila
  - Columna
- Para el tablero, se sigue utilizando la biblioteca graphics, la unica indicacion es que se use la estructura de datos mas adecuada, lo que me sugiere que es a libre eleccion, por lo tanto se uso un array bidimensional simple para el tablero

5.1. Regresando el modelo de ArrayList a array simple

- Como dije anteriormente, importamos las anterioes clases de videojuego, y en la clase que importamos, el tablero esta representado con un ArrayList
- Para trabajar mejor se hizo el pase de ArrayList a array simple
- Se logro eso, cambiando los lugares donde trabaja ArrayList

6. Actividad principal

A continuacion se cubriran los requerimientos principales de este laboratorio

6.1. Iniciando 2 ejercitos

En este punto, se inicial dos ejercitos con nombres autogenerados, para eso, se modifica el codigo de inicializacion de ejercitos, para permitir ingresar un parametro extra (Un numero entero que indica el numero de ejercito)

Listing 1: Codigo fuente, Videojuego.java

```
public static void main(String[] args){
   Soldado[] army1 = initializeArmy(0, true);
   Soldado[] army2 = initializeArmy(1, true);
...
  public static void displayArmy(Soldado[] army, String str){
    System.out.println("\n===== " + str + " =====");
    for(Soldado soldier : army){
      displaySoldier(soldier);
    }
  }
```

- Como vemos, los ejercitos se generaran con un numero que sera lo que contendra su nombre

7. Rúbricas

7.1. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio con el ítem correspondiente.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 2: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		18.5	