Universidad del Valle de Guatemala
CC2008 - Introducción a la Programación Orientada a Objetos
Semestre II, 2025
Ejercicio 1

Laboratorio 2: Arreglos y Excepciones

Análisis del Sistema (30 puntos)

1.1 Requisitos funcionales del sistema (5 pts)

Escribir los puntos exactos que debe cumplir tu sistema.

- El programa debe ser capaz de generar parejas de datos en un tablero que pueda mostrar.
- El programa debe ser capaz de identificar a dos jugadores y separarlos por turnos.
- El programa debe ser capaz de analizar códigos ASCII para generar emoticones.

1.2 Clases necesarias y su propósito (5 pts)

Clase	Propósito
Main	Instancia constructora del juego.
Controlador	Mantendrá el orden lógico del programa.
Consola	Se encargará de mostrar y responder al usuario.
Carta	Mantendrá la información de cada carta: símbolo, mostrado, emparejado.
Tablero	Organizará las cartas en forma cuadrada.
Jugador	Mantiene la información de los jugadores. Nombre, puntos y juegos ganados.

1.3 Atributos de cada clase (10 pts)

Main:

Atributo	Tipo de dato	Visibilidad	Propósito
memoria	Controlador	publico	Inicio del juego

Controlador:

Atributo	Tipo de dato	Visibilidad	Propósito
jugador1	Jugador	Private	Guardar información del jugador 1
jugador2	Jugador	Private	Guardar información del jugador 2
tablerito	Tablero	Private	Tablero del juego
consolita	Consola	Private final	

Consola:

Atributo	Tipo de dato	Visibilidad	Propósito
Sc	Scaner	Private final	Variable para realizar lectura en terminal.
Imp	System.out	Private final	Variable para facilitar impresión en terminal.

Carta:

Atributo	Tipo de dato	Visibilidad	Propósito
Símbolo	Char	Private final	Emoticono que representa la tarjeta.
Revelado	Boolean	Private	Si se ha seleccionado en el turno.
Emparejado	Boolean	Private	Si ya se determinó como pareja encontrada.

Universidad del Valle de Guatemala
CC2008 - Introducción a la Programación Orientada a Objetos
Semestre II, 2025
Ejercicio 1

Jugador:

Atributo	Tipo de dato	Visibilidad	Propósito
Nombre	String	Private final	Nombre del jugador asignado
Puntos	Int	Private	Puntos conseguidos en la ronda
Victorias	Int	Private	Rondas ganadas.

Tablero:

Atributo	Tipo de dato	Visibilidad	Propósito
Dimensiones	Int	Private	Obtener largo y ancho del tablero.
casillasTarjetas	Carta[][]	Private	Matriz del tablero. Con datos de dimensiones.
ParejasTotales	Int	Private	Total de parejas generadas.
ParjeasListas	Int	Private	Parejas que ya se encontraron.

1.4 Métodos de cada clase (10 pts)

Controlador:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
Void	Controlador	(Void)	Public	Constructor
Void	Turno	(Void)	Public	Se active cada turno y mantiene el control del juego
Void	Jugar	(Void)	Public	Se utiliza para iniciar un nuevo juego.

Consola:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
Void	Consola	(void)	Public	Constructor. Se establecen los atributos. El resto de métodos se encargan de la interacción con el usuario en la terminal.

Jugador:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
String	getNombre	(void)	Public	Modificar nombre
Void	setNombre	(String nombre)	Public	Obtener nombre
Int	getPuntos	(void)	Public	Modificar puntos
Void	setPuntos	(int modificador)	Public	Obtener puntos
Int	getVictorias	(void)	Public	Modificar victorias
Void	setVictorias	(int modificador)	Public	Obtener victorias

Universidad del Valle de Guatemala
CC2008 - Introducción a la Programación Orientada a Objetos
Semestre II, 2025
Ejercicio 1

Carta:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
Boolean	getSeleccionada	(Void)	Public	Revisar si ya fue seleccionada.
Void	Seleccionar	(Void)	Public	Seleccionar carta
Boolean	getEmparejada	(Void)	Public	Revisar si ya está emparejada.
Void	Emparejar	(Void)	Public	Emparejar carta
Char	getSimbolo	(Void)	Public	Obtener símbolo de la carta.
Void	quitarSeleccion	(Void)	Public	Quita selección de carta.

Tablero:

Atributo	Método	Parámetros : Tipo de dato	Visibilidad	Propósito
Int	getDimensiones	(Void)	Public	Obtener tamaño del tablero.
Tarjeta	getTarjeta	(int x, int y)	Public	Obtener tarjeta específica.
Int	getParejasTotales	(Void)	Public	Obtener parejas totales de tarjetas.
Int	getParjeasListas	(Void)	Public	Obtener parejas encontradas.
Void	setParjeasListas	(int modificador)	Public	Modificar parejas encontradas.

2. Diseño: Diagrama de Clases (30 puntos)

- Asegúrate de mostrar atributos y métodos con visibilidad (+, -).

- Indica relaciones entre clases (asociación, agregación, etc.).

- Incluye el driver program (Main).

Diagrama de clases aquí o adjunto en un archivo aparte.

3. Programa (40 puntos)

En cada archivo `.java`, asegurarse de incluir:

- Las clases necesarias.

- Uso adecuado de objetos.

Menú que debe implementar el driver program:

1. Nuevo comprador
2. Nueva solicitud de boletos
3. …
4. …
5. Salir

GitHub: colocar aquí la URL:

https://github.com/Hola2212/Lab2_Ar

reglos_y_Excepciones.git

Checklist antes de entregar

⬛ Está claro el análisis?

🟩 El diagrama tiene los elementos UML correctamente?

✅Subiste tu código a GitHub con todo lo necesario?