

Prueba de Resiliencia

Programación Orientada a Objetos – 26 de octubre 2024

Nombre		RUT	
Paralelo	() APaolini () MMoraga		

Antecedentes generales:

Puntaje total de la prueba/Puntos para nota aprobatoria (4.0)	100 puntos 60 puntos	Puntaje Obtenido	
Duración de la prueba	3 horas	Nota final	
Resultados de Aprendizaje a evaluar	1. Aplicar técnicas de ingeniería de software en la creación de software legible, mantenable y testeable. 2. Aplicar técnicas de programación orientada al objeto en la resolución de problemas. 3. Crear tipos de datos abstractos con bajo acoplamiento entre la implementación y su comportamiento que permitan la resolución de problemas. 4. Analizar las relaciones causa efecto de los procesos en estudio. 6. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.		
Fecha de entrega de resultados	10 noviembre 2024		

Instrucciones:

1. Esta evaluación tiene 4 páginas (incluyendo la portada). Compruebe que dispone de todas las páginas.
2. **Lea la prueba completamente DOS veces antes de hacer cualquier pregunta**
3. Durante la prueba no se puede utilizar: teléfonos móviles, apuntes. Está prohibido intentar conectarse a internet de cualquier manera (excepto a Campus Virtual, y solo para subir su solución). Si es sorprendido obtendrá la calificación mínima. Tampoco puede utilizar dispositivos de almacenamiento externos o cualquier otro dispositivo como relojes inteligentes, tables, audífonos, etc.
4. Una prueba respondida correctamente en un 60%, de acuerdo con las ponderaciones asignadas, corresponde a una nota 4,0.
5. La prueba es individual, cualquier sospecha de copia será calificada con la nota mínima y el caso será remitido al comité de ética.
6. En su espacio personal no debe haber nada más que hojas de papel en blanco, lápiz y goma.
7. El resto de sus implementos debe guardarlos dentro de su mochila/bolso y ésta debe posicionarse al frente debajo de la pizarra. Si leyó hasta este punto, felicidades, para saber que lo hizo dibuje un completo al final de esta página.
8. Los estudiantes a quienes se les compruebe falta de honestidad académica o cualquier otro acto contrario a las normas de permanencia universitaria o al espíritu universitario, serán sancionados, según sea la gravedad de la falta, con medidas desde la amonestación verbal hasta la suspensión o pérdida de la condición de estudiante. Los estudiantes expulsados no podrán volver a ingresar a ninguna carrera, programa o curso de la institución. El estudiante que incurriere en falta de honestidad, durante la realización de un proceso evaluativo, será calificado con la nota mínima 1,0.

Acepto las condiciones firmando: _____

Problema 1. Ruteo (30 puntos)

Rutee el siguiente código e indique la salida esperada. Dibuje la **situación final** de los objetos involucrados al terminar la ejecución. **No es necesario indicar todo el paso a paso.** Puede abreviar los strings más largos al dibujar.

```
1  public class Biblioteca {
2      private Libro[] libros = new Libro[10];
3
4      public void agregarLibro(Libro libro) {
5          for (int i = 0; i < libros.length; i++) {
6              if (libros[i] == null) {
7                  libros[i] = libro;
8                  System.out.println("Libro agregado: " + libro.getISBN());
9                  return; // termina el método sin devolver nada
10             }
11         }
12         System.out.println("No hay espacio para agregar más libros.");
13     }
14
15     public void prestarLibro(String isbn) {
16         for (Libro libro : libros) {
17             if (libro != null && libro.getISBN().equals(isbn)) {
18                 libro.prestar();
19                 return;
20             }
21         }
22         System.out.println("Libro con ISBN " + isbn + " no encontrado.");
23     }
24
25     public void mostrarLibros() {
26         System.out.println("Catálogo de libros:");
27         for (Libro libro : libros) {
28             if (libro != null) {
29                 System.out.println(libro);
30             }
31         }
32     }
33 }
```

Ingeniería en Tecnologías de Información
Ingeniería Civil en Computación e Informática
Ingeniería Civil Industrial

```
1 public class Main {  
2  
3     public static void main(String[] args) {  
4         Biblioteca biblioteca = new Biblioteca();  
5  
6         Libro libro1 = new Libro("1984", "George Orwell", "4935");  
7         Libro libro2 = new Libro("Cien Años de Soledad", "Gabriel García Márquez", "3287");  
8         Libro libro3 = new Libro("El Quijote", "Miguel de Cervantes", "0049");  
9  
10        biblioteca.agregarLibro(libro1);  
11        biblioteca.agregarLibro(libro2);  
12        biblioteca.agregarLibro(libro3);  
13  
14        biblioteca.mostrarLibros();  
15  
16        biblioteca.prestarLibro("4935");  
17        biblioteca.prestarLibro("3287");  
18        biblioteca.prestarLibro("4935");  
19        biblioteca.prestarLibro("4900");  
20        biblioteca.agregarLibro(new Libro("Crimen y castigo", "Fiódor Dostoyevski", "6975"));  
21        biblioteca.mostrarLibros();  
22    }  
23}  
24  
25}
```

```
1 public class Libro {  
2  
3     private String titulo;  
4     private String autor;  
5     private String isbn;  
6     private boolean prestado;  
7  
8     public Libro(String titulo, String autor, String isbn) {  
9         this.titulo = titulo;  
10        this.autor = autor;  
11        this.isbn = isbn;  
12        this.prestado = false;  
13    }  
14  
15    public String getISBN() {  
16        return isbn;  
17    }  
18  
19    public boolean estaPrestado() {  
20        return prestado;  
21    }  
22  
23    public void prestar() {  
24        if (!prestado) {  
25            prestado = true;  
26            System.out.println("Préstamo aprobado: " + titulo);  
27        } else {  
28            System.out.println("Préstamo rechazado: " + titulo + " ya está prestado.");  
29        }  
30    }  
31  
32    @Override  
33    public String toString() {  
34        if (prestado)  
35            return titulo + ", " + autor + ", ISBN=" + isbn + ", " + "Prestado";  
36        else  
37            return titulo + ", " + autor + ", ISBN=" + isbn + ", " + "Disponible";  
38    }  
39}
```

Ingeniería en Tecnologías de Información

Ingeniería Civil en Computación e Informática

Ingeniería Civil Industrial

Problema 2. Predicción de Partidos (70 puntos)

En una liga de fútbol internacional, cada equipo está compuesto por jugadores que poseen diversas habilidades y estadísticas. La liga desea implementar un sistema para predecir el posible ganador de cada partido, basado en las estadísticas de los jugadores y el rendimiento agregado de cada equipo.

Descripción del Sistema

Cada equipo en la liga cuenta con un nombre, un ID único y está asociado a un país. Los equipos están conformados por un grupo de jugadores titulares (máximo 11), quienes participan activamente en los partidos y contribuyen al rendimiento general del equipo. No se consideran jugadores suplentes o cambios en este sistema de predicción.

Cada jugador tiene un nombre, un número de camiseta único y está vinculado a un equipo específico a través del ID de dicho equipo. Además, cada jugador posee atributos de rendimiento en el campo, como velocidad, resistencia, precisión, defensa y ataque, los cuales se valoran en una escala de 1 a 100. La liga cuenta con un calendario de partidos programados en los que se enfrentan dos equipos en cada encuentro. El objetivo del sistema es predecir el posible ganador de cada partido utilizando las estadísticas agregadas de los equipos.

Objetivos del Sistema:

1. Procesamiento de Información de Equipos

- Leer y procesar los datos de cada equipo desde un archivo `equipos.txt`.

2. Procesamiento de Información de Jugadores

- Leer y procesar la información de cada jugador desde un archivo `jugadores.txt`, asignándoles a sus respectivos equipos mediante el ID del equipo.

3. Cálculo de Puntuaciones de Equipos

- Calcular una puntuación total para cada equipo sumando las estadísticas de todos los jugadores titulares. Esta puntuación representará el rendimiento potencial del equipo en un partido.

4. Predicción de Resultados de Partidos

- Leer el calendario de partidos desde un archivo `partidos.txt` y, basándose en las puntuaciones de los equipos, predecir el equipo con mayor probabilidad de ganar cada encuentro.

5. Estadísticas por País

- Calcular y mostrar el promedio de puntuación de equipos por país, permitiendo conocer los países con mejor rendimiento global y dónde podrían encontrarse los mejores jugadores.

6. Listado de Jugadores Destacados

- Mostrar a los jugadores con mejores estadísticas, ordenados de mayor a menor, para permitir que los equipos interesados utilicen esta información al considerar nuevas contrataciones.

7. Eliminación de Jugadores Duplicados

- Validar que no existan jugadores duplicados. En caso de encontrar duplicados, ambos registros deben eliminarse. Esta condición será evaluada como parte del sistema.

Estructura de los Archivos

Cuando se inicia el programa se debe cargar el archivo de inventario de los equipos, los jugadores de cada equipo y de los partidos para realizar las predicciones. En el archivo `equipos.txt` cada línea del archivo sigue la estructura:

Archivo	Explicación
1,Real Madrid,España	id: (int) Identificador único del equipo.
2,Liverpool,Inglaterra	Nombre: (String): Nombre del equipo.
3,Boca Juniors,Argentina	País: (String): País al que representa
:	

Nota: Los nombres de equipo se pueden repetir, pero son de un país diferente y tienen un id único que los diferencie.

Ingeniería en Tecnologías de Información

Ingeniería Civil en Computación e Informática

Ingeniería Civil Industrial

El archivo **jugadores.txt** contiene la información de cada jugador, el cual sigue la siguiente estructura:

Archivo	Explicación
Sergio Ramos,4,1,80,85,70,90,60	id: (int) Identificador único del equipo.
Mohamed Salah,11,2,95,90,85,70,95	numeroCamiseta: (int): Número de camiseta del jugador.
Carlos Tevez,10,3,85,80,80,75,90	idEquipo: (int): ID del equipo al que pertenece.
.	velocidad : Estadísticas (int, valores de 1 a 100)
.	resistencia: Estadísticas (int, valores de 1 a 100)
.	precision: Estadísticas (int, valores de 1 a 100)
.	defensa: Estadísticas (int, valores de 1 a 100)
.	ataque: Estadísticas (int, valores de 1 a 100)

Los partidos para realizar predicciones deben leerse desde el archivo **partidos.txt** y contiene la siguiente estructura:

Archivo	Explicación
1,2	idEquipoLocal,idEquipoVisitante
2,3	idEquipoLocal: (int): ID del equipo al que pertenece.
3,1	idEquipoVisitante: (int): ID del equipo al que pertenece.
.	
.	
.	

Debe entregar:

- Modelo de Dominio (10%)
- Diagrama de clases (15%)
- Código Java (75%)

Consideraciones:

- Debe usar orientación a objetos.
- No se deben utilizar ciclos dentro de ciclos. Use funciones para hacer la vida más fácil.
- El código fuente debe exportarlo en un solo archivo .zip y subirlo a Campus Virtual.
- Los archivos .txt de ejemplo se encuentran en Campus Virtual.
- Los diagramas de clases debe escribirlos en papel y entregarlos junto a la prueba.
- Hojas sin nombre no se revisarán.
- La cantidad de partidos no se sabe, pero se sobredimensiona a 100 partidos.
- De los jugadores no se sabe cuántos son por lo tanto se sobredimensiona a 1000.
- Validar que los jugadores no se repitan, en caso de encontrar duplicados, ambos registros deben eliminarse. Esta condición será evaluada como parte del sistema.
- Debe usar referencias
- Puede usar Lists.
- Si lee esto quiere decir que leyó las consideraciones, dibuje una bola disco al final de la página.