Diseño de Estructuras de Datos

PR2: Práctica 2

Enunciado

A continuación os presentamos el enunciado de la segunda parte de la práctica (PR2) del curso. La práctica consiste en un ejercicio de programación evolutivo sobre el primer ejercicio práctico (PR1). Es **OBLIGATORIO** que la codificación de este ejercicio se base en las estructuras de datos definidas en la solución oficial de la PEC1 y la PEC2; y utilizar como punto de partida la codificación de la PR1 (solución oficial). **NO utilizéis vuestra solución**. Si es necesaria alguna estructura adicional para alguna de las funcionalidades se pueden añadir justificando esta decisión.

Os recordamos que para superar la asignatura es necesario obtener una nota mínima de 5 de prácticas (PR0, PR1 y PR2). Por otro lado, esta práctica debe resolverse individualmente y todo indicio de copia será notificado a los responsables de los estudios para que tomen las medidas oportunas.

Descripción de la PR2 a realizar

En la PR2 trabajaremos básicamente con las siguientes entidades y conceptos:

- **Player**: Jugador que participa en los eventos deportivos
- **OrganizingEntity**: Organización que propone eventos deportivos
- **File:** Expediente que permite, en caso de validación, iniciar un nuevo evento deportivo
- **SportEvent:** Evento deportivo
- **Rating:** Valoración de la actividad cultural
- **Worker:** Trabajador del club deportivo
- **Role:** Rol de los trabajadores
- **Attendee:** Asistentes a eventos deportivos

Funcionalidades

Las funcionalidades requeridas para la PR2 son:

- addRole(roleId, description)
- addWorker(dni, name, surname, birthday, roleId)
- assignWorker(dni, eventId)
- getWorkersByEvent(eventId): Iterador
- getWorkersByRole(roleId): Iterador

- getLevel(playerId): Level
- getSubtitutes(eventId): Iterador
- addAttender(phone, name, eventId)
- getAttender(phone, eventId): Attender
- getAttenders(eventId): Iterador
- best5OrganizingEntities(): Iterador
- bestEvent(): Event

Operaciones opcionales:

- addFollower(playerId, playerFollowerId)
- getFollowers(playerId): Iterador
- getFollowings(playerId): Iterador
- recommendations(playerId): Iterador
- getPosts(playerId): Iterador

Adicionalmente, se han definido nuevas operaciones que permiten inspeccionar la estructura de datos y validar los juegos de pruebas:

- numPlayers(): int
- public int numOrganizingEntities(): int
- public int numFiles(): int
- public int numRejectedFiles(): int
- int numPendingFiles(): int
- int numSportEvents(): int
- int numSportEventsByPlayer(String playerId): int
- int numSportEventsByOrganizingEntity(int orgId): int
- getPlayer(String playerId): Player
- getSportEvent(String eventId): SportEvent
- getOrganizingEntity(int id): OrganizingEntity
- currentFile(): File
- numRoles(): int
- getRole(roleId): Role
- numWorkers(): int
- getWorker(): Worker
- numWorkersBySportEvent(String sportEventId):int
- numWorkersByRole(String roleId): int
- numRatings(String playerId): int
- numAttenders(String sportEventId): int
- numFollowers(String idPlayer): int
- numFollowings(String idPlayer): int

Todas las operaciones requeridas en la PR2 están definidas en la interfaz **SportEvents4Club** que se proporciona junto con este documento.

Puesta en marcha del proyecto

Tal como se indica en el apartado de Recursos, se proporciona un proyecto base para realizar el ejercicio. En concreto:

- src/main/java/uoc.ds.pr.**SportEvents4Club.java**: Interfaz que especifica las operaciones del TAD a implementar
- src/test/java/uoc.ds.pr.**FactorySportEvents4Club.java**: Clase que implementa el patrón de diseño factoría y que inicializa las estructuras de datos con valores iniciales para **jugadores, organizaciones, ficheros, roles, trabajadores, …**
- **SportEvents4ClubPR1Test.java, SportEvents4ClubPR2Test.java** y **SportEvents4ClubPRjava**: Clases de test del TAD **SportEvents4Club**
- **lib/DSLib-2.1.1.jar**: Versión 2.1.1 de la DSLib. Notad que no es la misma versión que utilizásteis para desarrollar la PR1 (en este caso, utilizamos la versión previa 2.1.0).
- Clases pendientes de implementar:
    - **SportEvents4ClubImpl** y las entidades del modelo definidas
    - Adicionalmente se deberán implementar todas las excepciones definidas en el interfaz que deben heredar de la clase **DSDExcepcion** que se proporciona en el enunciado

Instalación

Los pasos principales para poner en marcha el proyecto de esta práctica son los siguientes:

- descargar, descomprimir el .zip con el enunciado de la PR2 e importar el proyecto en el IDE correspondiente
- implementación de las partes pendientes de implementar.
- ejecutar los JUnit tests src/test/java

NOTA: Contactad con el profesorado del aula de Laboratorio de la asignatura para cualquier duda en este punto y en otros relacionados con la codificación de esta práctica.

Planificación

La planificación que os proponemos para la **PR2** es la siguiente:

1. (22/12-23/12). Análisis de la solución de la PEC1, PEC2 y PR1, e identificación de los TAD de la librería.

2. (23/12-26/12). Análisis del juego de pruebas y validación de que las estructuras de datos propuestas en la PEC1 y PEC2 son suficientes para cumplir con los requerimientos del juego de pruebas. En caso de que exista alguna modificación habrá que indicarlo en el documento de texto de la entrega. Con esta acción estamos siguiendo una metodología de desarrollo dirigido por las pruebas (*Test-driven development*,TDD).

3. (26/12-16/1). Implementación del gestor con las operaciones definidas por la interfaz proporcionada.

   NOTA: Para la implementación de las tests relacionados con la red social (**SportEvents4ClubPR2TestPlus)** se recomienda analizar previamente el test unitario **SocialNetworkWithUndirectedGraphTest** de la librería DSLib que puede ser consultado a través del repositorio GIT.

4. (16/1-25/1). Pruebas, validación de los juegos de pruebas y actualización en caso de que sea necesario.

Formato de entrega

La entrega hay que hacerla en un archivo comprimido (ZIP), a través del espacio de "Entrega y registro de EC", organizado de la siguiente manera:

- Un documento de texto (leeme.txt) indicando el alcance de la entrega, modificaciones y / o actualizaciones realizadas sobre el diseño inicialmente propuesto (solución oficial de la PEC2) con su justificación en el caso de que haya sido necesaria alguna estructura de datos adicional, problemas y comentarios adicionales.
- Una presentación de las pruebas ejecutadas. Recordad que no tenéis por qué quedaros únicamente con el juego de pruebas que os proporcionamos: si lo consideráis oportuno, podéis ampliar dicho juego de pruebas. Se deben incluir directamente en el ZIP.
- El proyecto con sus fuentes: código (*.java) manteniendo la estructura del proyecto (carpeta src/main, src/test y estructura de paquetes. NO ADJUNTAR ni archivos *.class ni la biblioteca de TADs de la asignatura en el ZIP. A efectos prácticos podéis comprimir con un zip el proyecto sin los binarios.