Mateo Valdés

17 de octubre de 2019

Algoritmos y programación II

Laboratorio 4

1. Requerimientos funcionales:

RF1: Registrar espectadores y participantes a partir de un archivo

RF2: Buscar a un espectador por ID

RF3: Buscar a un participante por ID

RF4: Mostrar los datos de los espectadores de un país determinado

RF5: Mostrar los datos de los participantes de un país determinado

Especificación de requerimientos funcionales:

Nombre	RF1: Registrar espectadores y participantes a partir de un archivo
Resumen	El sistema obtiene la ruta relativa de un archivo de texto con los datos y partir de ese archivo de texto registra los espectadores y los participantes
Entrada	La ruta del archivo con la información
Salida	Los espectadores y los participantes han sido registrados exitosamente

Nombre	RF2: Buscar a un espectador por ID
Resumen	El sistema permite buscar a un espectador dado su ID
Entrada	El ID del espectador
Salida	Los datos del espectador que ha encontrado

Nombre	RF3: Buscar a un participante por ID
Resumen	El sistema permite buscar a un participante dado su ID
Entrada	El ID del participante
Salida	Los datos del participante que ha encontrado

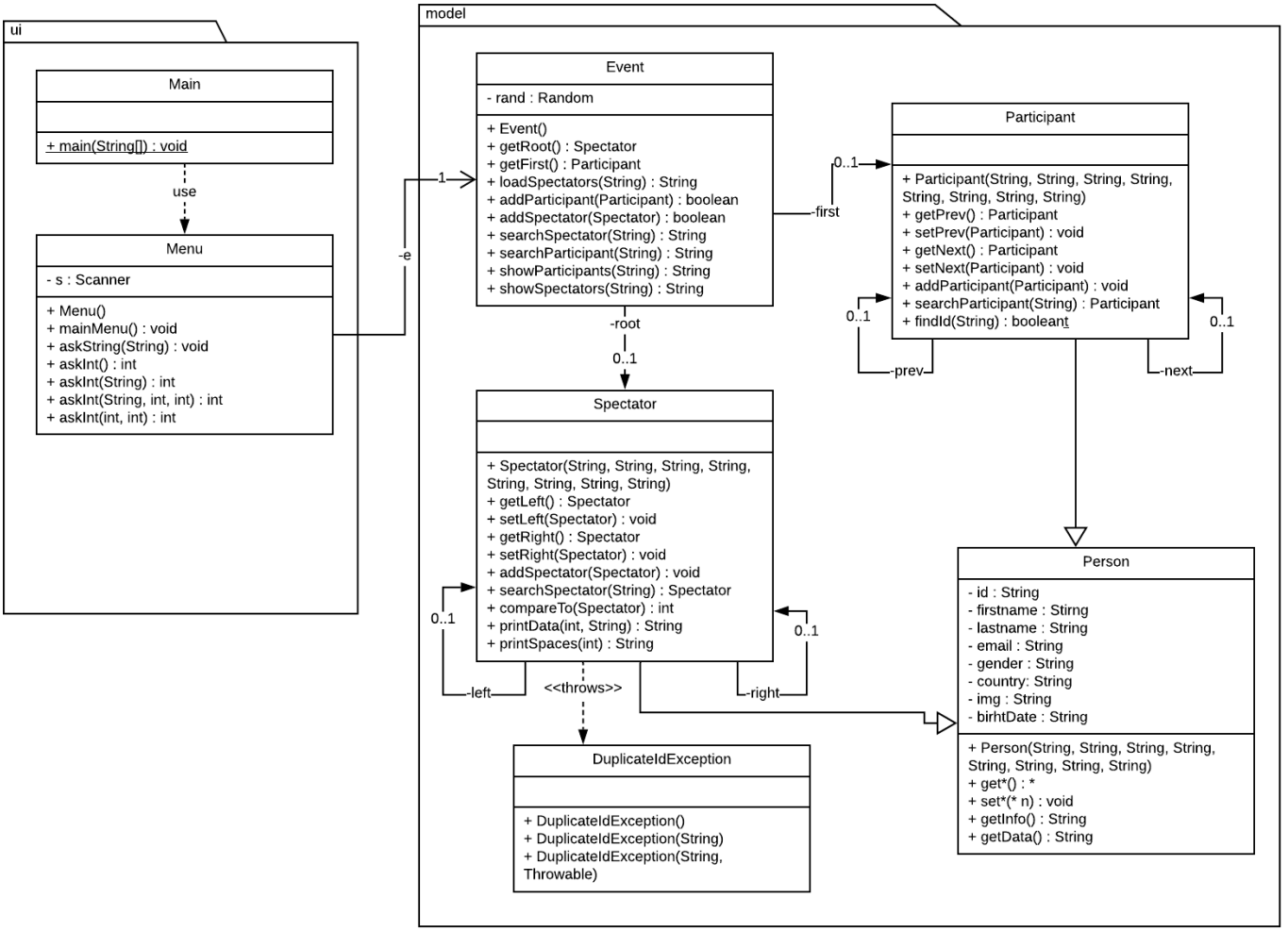
Nombre	RF4: Mostrar los datos de los espectadores de un país determinado
Resumen	El sistema muestra los datos de los espectadores de un país que el usuario digita
Entrada	El país de los espectadores
Salida	Los datos de todos los espectadores del país deseado

Nombre	RF5: Mostrar los datos de los participantes de un país determinado
Resumen	El sistema muestra los datos de los participantes de un país que el usuario digita
Entrada	El país de los participantes
Salida	Los datos de todos los participantes del país deseado

Requerimientos no funcionales:

- El programa debe leer un archivo plano de tipo .txt para obtener los datos de los espectadores y los participantes
- Los datos de los espectadores deben ser guardados en un árbol binario de búsqueda.
- Los datos de los participantes deben ser guardados en una lista doblemente enlazada
- Se deben hacer varias pruebas en JUnit para asegurarse que le programa funciona satisfactoriamente
- Los métodos de los árboles binarios de búsqueda deben ser recursivos

2. Diagrama de clase del modelo e interfaz (no generado automáticamente)



3. Diseño de casos de pruebas unitarias

Clase: Event

Escenarios:

loadScene1(): Se crea un evento y un arreglo de espectadores de tamaño 5. A ese arreglo se le agregan cinco espectadores con las siguientes características:

- Espectador #1:
  - ID: 32-1234
  - Nombre: Johan
  - Apellido: Giraldo
  - Email: jg@gmail.com
  - Sexo: Male

    - País: Colombia
    - Imagen: johan.jpg
    - Fecha de nacimiento: 1999/12/25
- Espectador #2:
    - ID: 45-5134
    - Nombre: Esteban
    - Apellido: Ariza
    - Email: ea@gmail.com
    - Sexo: Male
    - País: Japan
    - Imagen: img2.jpg
    - Fecha de nacimiento: 1995/11/21
- Espectador #3:
    - ID: 12-1654
    - Nombre: Esteban
    - Apellido: Yusunguaira
    - Email: ey@gmail.com
    - Sexo: Female
    - País: China
    - Imagen: rere.jpg
    - Fecha de nacimiento: 1992/07/06
- Espectador #4:
    - ID: 54-5234
    - Nombre: Jhun
    - Apellido: Kim
    - Email: jk@gmail.com
    - Sexo: Male
    - País: South Korea
    - Imagen: kim.jpg
    - Fecha de nacimiento: 2000/01/10
- Espectador #5:
    - ID: 04-1834
    - Nombre: Juan
    - Apellido: Ossa
    - Email: jo@gmail.com
    - Sexo: Female
    - País: Haiti
    - Imagen: ossa.jpg

Método	Escenario	Valores de entrada	Resultado

addSpectator()	loadScene1()	-	True si la los cinco espectadores se agregan de una manera ordenada por el ID al árbol binario de búsqueda con el espectador #1 siendo la raíz dentro del evento (porque es el primero en ser agregado).

Clase: Spectator

Escenarios:

loadScene1(): Se crea un arreglo de espectadores de tamaño 5. A ese arreglo se le agregan cinco espectadores con las siguientes características:

- Espectador #1:
  - ID: 32-1234
  - Nombre: Johan
  - Apellido: Giraldo
  - Email: jg@gmail.com
  - Sexo: Male
  - País: Colombia
  - Imagen: johan.jpg
  - Fecha de nacimiento: 1999/12/25
- Espectador #2:
  - ID: 45-5134
  - Nombre: Esteban
  - Apellido: Ariza
  - Email: ea@gmail.com
  - Sexo: Male
  - País: Japan
  - Imagen: img2.jpg
  - Fecha de nacimiento: 1995/11/21
- Espectador #3:
  - ID: 12-1654
  - Nombre: Esteban
  - Apellido: Yusunguaira
  - Email: ey@gmail.com
  - Sexo: Female
  - País: China
  - Imagen: rere.jpg
  - Fecha de nacimiento: 1992/07/06
- Espectador #4:
  - ID: 54-5234

    - Nombre: Jhun
    - Apellido: Kim
    - Email: jk@gmail.com
    - Sexo: Male
    - País: South Korea
    - Imagen: kim.jpg
    - Fecha de nacimiento: 2000/01/10
- Espectador #5:
    - ID: 04-1834
    - Nombre: Juan
    - Apellido: Ossa
    - Email: jo@gmail.com
    - Sexo: Female
    - País: Haiti
    - Imagen: ossa.jpg
    - Fecha de nacimiento: 1997/04/06

Método	Escenario	Valores de entrada	Resultado
addSpectator()	loadScene1()	-	True si los cinco espectadores se agregan de una manera ordenada por el ID al árbol binario de búsqueda y el método lanza una excepción cuando se intenta agregar un espectador con un ID igual al de un espectador que ya está en el árbol binario de búsqueda
searchSpectator()	loadScene1()	-	True si retorna cada uno de los espectadores después de buscarlos por sus respectivos IDs

LINK DE GITHUB: https://github.com/mavaldot/Voleibol