

Primer parcial de Estructuras de datos/Algoritmos I

Año 2024

Aclaración importante: En ningún ejercicio está permitido eliminar atributos existentes en las clases, ni modificar los perfiles de los métodos provistos, ni eliminar métodos. Sí está permitido agregar los atributos y las funciones auxiliares que desee. Puede dar una solución sin usar los atributos existentes (aunque esto probablemente reduzca un poco la cantidad de tests que pasen). Tampoco debe agregar nuevas clases que no sean requeridas en la consigna.

Ejercicio 1. Implementar todos los métodos de la clase `MusicOrganizer.java` de manera que satisfagan las especificaciones provistas.

La clase `MusicOrganizerTest` tiene algunos tests que pueden servir de guía para saber el grado en que su implementación satisface las especificaciones.

Nota: En la carpeta `docs/ArrayList.pdf` se encuentra la documentación de la clase `java.util.ArrayList`.

Ejercicio 2. Implementar todos los métodos de la clase `MusicOrganizer.java` de manera que satisfagan las especificaciones provistas, pero esta vez el código debe usar su propia implementación de listas con arreglos. Escriba su implementación de listas en la clase `ListArray.java`. `ListArray` debe implementar los métodos de la interfaz `List`, y el código de `MusicOrganizer` debe estar programado respecto de esta interfaz. Específicamente, debe implementar los métodos no implementados de la clase `ListArray.java`: `add(T e)`, `resize(int capacity)`, `get(int index)`, `remove(int index)`, `contains(T e)`, y la función de abstracción en el método `toString()`.

Las clases `MusicOrganizerTest` y `ListArrayTest` tienen algunos tests que pueden servir de guía para saber el grado en que su implementación satisface las especificaciones.

Nota: No puede importar ninguna implementación de las provistas por Java para resolver este ejercicio. Si utiliza una estructura de Java (por ejemplo, haciendo un `import`) tendrá 0 puntos en el ejercicio.

Ejercicio 3. Implemente pilas con listas simplemente encadenadas en la clase `LinkedStack` (todos los métodos menos `iterator`). Estos métodos deben respetar las especificaciones del TAD provistas en la interfaz `Stack`.

La clase `Utils` contiene un cliente de `Stack` que puede ayudar a evaluar el funcionamiento correcto de su implementación.

Las clases `UtilsTest` y `LinkedStackTest` tienen algunos tests que pueden servir de guía para saber el grado en que su implementación satisface las especificaciones.

Nota: No puede importar ninguna implementación de las provistas por Java para resolver este ejercicio. Si utiliza una estructura de Java (por ejemplo, haciendo un import) tendrá 0 puntos en el ejercicio.

Ejercicio 4. Implemente los métodos `inscribirEstudiante(Student newStudent)`, `getEstudiantesGrado()` y `getEstudiantesPosgrado()` de `LabClass.java`, las clases `EstudianteGrado.java` y `EstudiantePosgrado.java`, y el método `equals` de `Student.java`, de modo que satisfagan sus especificaciones.

Las clases `LabClassTest` y `StudentTest` tienen algunos tests que pueden servir de guía para saber el grado en que su implementación satisface las especificaciones.