Ing. de Sistemas e Informática
Estructuras de Datos y Algoritmos
Jorge Mario Londoño P.

Taller 1

Tipos de datos abstractos (ADT) y uso de bibliotecas (APIs)

Introducción

Una biblioteca de funciones encapsula tipos de datos abstractos y algoritmos de uso frecuente en el desarrollo de software.

Los autores de el texto guía ofrecen un API para facilitar operaciones frecuentemente utilizadas. La biblioteca de funciones se encuentra disponible en la siguiente dirección:

http://algs4.cs.princeton.edu/code/

Allí están disponibles el código fuente, la documentación y archivos ejemplos para los distintos algoritmos. La biblioteca entera está disponible como un archivo JAR (algs4.jar), el cual se debe incluir en el CLASSPATH del proyecto (ver instrucciones en la misma página).

Para efectos de esta práctica nos interesan 5 clases incluidas en esta biblioteca:

`StdIn` : Funciones para lectura de datos por consola

`StdOut` : Funciones para escritura de datos en consola

`In` : Funciones para leer datos desde archivos o URLs

`Out` : Funciones para escribir datos en archivos

`StdDraw` : Funciones para hacer gráficas

`StdRandom` : Funciones para generar números aleatorios

`StdStats` : Cálculos estadísticos básicos

Otra biblioteca de uso frecuente está incluida en el API de Java es la clase `Arrays`, que nos permite hacer varias cosas utilizando arreglos de datos: Ordenarlos, convertirlos a `String`, hacer búsquedas, etc. La clase está documentada aquí: `java.util.Arrays`.

Ejercicio a desarrollar

Se desea implementar una calculadora de vectores. Para tal fin, se require un ADT Vector que represente vectores en el espacio (se puede generalizar para vectores en $\mathbb{R}^n$). Queremos que la calculadora ofrezca las principales operaciones de vectores:

Operación	**Operador**	**Fórmula**
Suma	+	$(\ldots,a_i+b_i,\ldots)$
Resta	-	$(\ldots,a_i-b_i,\ldots)$
Producto punto	*	$\sum_{i=1}^{n} a_i b_i$
Producto cruz	x	$\begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$
Norma del vector	||	$\sqrt{\sum_{i=1}^{n} a_i^2}$

La implementación del ADT debe constar de:

1. La clase `Vector` que encapsula un vector y permite realizar las operaciones básicas descritas en la tabla. Asegurarse de proteger la representación.
2. Un método estático main en la clase `Vector` que realice al menos una prueba unitaria por cada operación. No hacer entradas/salidas, en caso de fallar una prueba debe arrojar la correspondiente exception (usar `assert`).
3. Implementar una función de biblioteca `parseVector`, que toma como entrada un String y devuelve la instancia de `Vector` correspondiente.
   ```
   static Vector parseVector(String s)
   ```
4. Implementar los métodos `equals`, `toString` heredados de la clase Object.
5. Implementar una aplicación cliente `CalculadoraVectores` basada en el ejemplo de clase de la calculadora RPN, que haga uso de las bibliotecas del texto para entradas y salidas (`StdIn`, `StdOut`) y que le permita al usuario realizar las operaciones definidas con números complejos.

Entregables:

Enviar solo el código fuente como un archivo comprimido (se aceptan .zip, .rar, .7z, .tgz). No incluir dentro del comprimido la biblioteca algs4.jar.

Seguir el estándar de nombres: Practica1-<NombreApellido1>-<NombreApellido2>.zip.

Se puede realizar el trabajo en equipos de máximo 2 personas.