Guion de Prácticas

Estructuras de Datos [ED]

Curso 2021 – 2022

Proyecto 4: Tablas Hash

Objetivos de Aprendizaje

Al final de la unidad el alumno será capaz de:

1. Implementar tablas hash cerradas basándose en una clase abstracta
2. Implementar tablas hash abiertas basándose en una clase abstracta

Proyecto

Durante la unidad, el alumno creará un proyecto compuesto por los siguientes elementos:

- La clase **HashNode** que implementa un nodo de una tabla hash cualquiera
- La clase abstracta **AbstractHash** donde se definen métodos que se implementarán de distinta forma dependiendo de si se trata de tablas hash cerradas o abiertas y donde se implementa otros que serán comunes a ambas implementaciones.
- La clase **ClosedHashTable** que implementa una tabla hash cerrada cuyos elementos con HashNode.
- La clase **OpenHashTable** que implementa una tabla hash abierta cuyos elementos son HashNode (***opcional***)

Cada una de las clases contendrá los siguientes métodos. Se especifica la cabecera de los métodos para la clase genérica.

Clase HashNode

Crear una clase que defina las propiedades de un nodo de una tabla hash, así como los métodos para gestionar dichas propiedades.

- ***public HashNode()***
  Constructor vacío, asigna a la propiedad info de la clase el valor ***null*** y pone el estado a ***VACIO***
- ***public T getInfo()***
  Devuelve el contenido de la propiedad info de la tabla hash
- ***public void setInfo(T elemento)***
  Asigna un valor a la propiedad info y el estado pasa a ser ***LLENO***
- ***public void remove()***
  Cambia el estado del nodo de la tabla hash para que sea ***BORRADO*** pero no le asigna ***null*** a info
- ***public int getStatus()***
  Devuelve el estado del nodo de la tabla hash
- ***public String toString()***
  Muestra el contenido del nodo entre llaves, {_E_} si el estado es vacío y {_D_} si el estado es borrado

Clase abstracta AbstractHash

Crear una clase abstracta que defina los métodos de cualquier cola de cualquier tabla hash y otros implementados válidos para cualquier tabla hash.

- ***public int getNumOfElems()***
  Devuelve el número de elementos insertados en la tabla hash
- ***public int getSize()***
  Devuelve el tamaño de la tabla hash
- ***public int add(T elemento)***
  Añade un nuevo elemento a la tabla hash. Devuelve 0 si lo inserta correctamente, -1 si no lo puede insertar y -2 si el elemento a insertar es ***null***. No se admiten elementos repetidos
- ***public T find(T elemento)***
  Busca en la tabla hash, el elemento que se le pasa como parámetro. Devuelve el elemento encontrado si lo encuentra y ***null*** en caso contrario
- ***public int remove(T elemento)***
  Elimina de la tabla hash, el elemento que se le pasa como parámetro. Devuelve 0 si lo borra correctamente, -1 si no se puede borrar y -2 si es ***null***
- ***public void reDispension()***
  Aumenta el tamaño de la tabla hash
- ***public void inverseReDispension()***
  Disminuye el tamaño de la tabla hash

- ***public String toString()***
  Construye una cadena con el contenido de la tabla hash, así como su tamaño y el número de elementos de la tabla
  Ejemplo:
  {10};{_E_};{_E_};{8};{_E_};[Size: 5 Num.Elems.: 2]

Los siguientes métodos son iguales para cualquier implementación de una tabla hash, luego se implementarán en la clase abstracta.

- ***protected int fHash(T elemento)***
  Calcula la posición del elemento dentro de la tabla hash
- ***protected boolean isPositivePrime(int numero)***
  Devuelve ***true*** si el valor que se le pasa como parámetro es primo y ***false*** en caso contrario. Los números negativos no los consideraremos primos
- ***protected int nextPrimeNumber(int numero)***
  Devuelve el siguiente número primo mayor que el número pue se pasa como parámetro
- ***protected int previousPrimeNumber(int numero)***
  Devuelve el siguiente número primo menor que el número pue se pasa como parámetro. Nunca dejaremos que sea menor que 3

Clase ClosedHashTable

Implementar los métodos definidos en la clase abstracta para tablas hash cerradas. Se permiten claves repetidas.

Clase OpenHashTable

Implementar los métodos definidos en la clase abstracta para tablas hash abiertas.

Consideraciones finales

- Todas las clases, propiedades y métodos escritas en el proyecto debe ser documentadas correctamente por el alumno utilizando **JavaDoc**.
- Todos los métodos diseñados deben ser evaluados por el alumno mediante una exhaustiva batería de pruebas positivas y pruebas negativas haciendo uso de **JUnit**.

Fecha de Entrega

Una vez completado el proyecto, deberá subirse al Campus Virtual en las fechas que se designen. Pasado ese período no se aceptará la entrega del proyecto de ningún modo.