**Tistea Stefan**

**Massaoudi Omar**

**Overview**

The DatabaseEngine class implements a B+ tree to store and manage student records efficiently. It provides functionalities to insert, search, update, and delete student records based on various attributes.

**Components**

1. **Student Class**:
   * Represents a student with attributes like ID, name, total score, exam score, and notes/lab information.
   * Generates a unique ID for each student.
2. **DictionaryPair Class**:
   * A key-value pair used in leaf nodes, where the key is the student ID and the value is the student object.
3. **Node Classes**:
   * **Node**: Abstract base class for InternalNode and LeafNode.
   * **InternalNode**: Represents internal nodes in the B+ tree, containing keys and pointers to child nodes.
   * **LeafNode**: Represents leaf nodes in the B+ tree, containing dictionary pairs and pointers to sibling leaf nodes.

**Core Methods**

1. **Insertion**:
   * Adds a new student record to the tree. If the tree is empty, it creates the first leaf node.
   * If the target leaf node is full, it handles overflow by splitting nodes as necessary.
2. **Search**:
   * **By ID**: Finds a student by their unique ID.
   * **By ID Range**: Retrieves a list of students within a specified range of IDs.
   * **By Name**: Finds students with a specified name.
   * **By Total Score**: Finds students with a specified total score.
   * **By Exam Score**: Finds students with a specified exam score.
   * **By Notes/Lab**: Finds students with specified notes or lab information.
3. **Update**:
   * Modifies the attributes of an existing student record identified by their ID.
4. **Deletion**:
   * Removes a student record from the tree by their ID.
   * Handles node deficiencies by merging or redistributing nodes as necessary.

**Utility Methods**

* **isEmpty**: Checks if the B+ tree is empty.
* **findLeafNode**: Locates the appropriate leaf node for a given key.
* **binarySearch**: Performs a binary search within a node's dictionary pairs.
* **linearNullSearch**: Finds the first null position in an array, used for determining the number of elements in nodes.

**Main Method**

* Provides a command-line interface for interacting with the DatabaseEngine.
* Offers a menu for users to insert, search, update, and delete student records or exit the program.
* Uses a scanner to read user inputs and perform corresponding actions on the database.

**Usage**

The DatabaseEngine class is designed to efficiently manage large collections of student records, allowing quick insertion, retrieval, and modification of data. The B+ tree structure ensures that search and update operations are performed in logarithmic time, making it suitable for applications requiring high performance and scalability.