CS 5040 (Dr. Shaffer): Project 1: try #9 (10.0/100.0)

src/HashTable.java

View: Overall summary of results Go

```
import java.io.PrintWriter;
import java.util.*;

public class HashTable {
```

Error [Checkstyle]: -2
The Javadoc for this class or interface is missing. All visible (i.e. not private) classes and interfaces must be docume

```
    private static final double LOAD_FACTOR_THRESHOLD = 0.5;

    private Record[] table;
    private int size;
    private int memoryPoolSize;
    private int[] freeBlocks;
    private PrintWriter writer;

    public HashTable(int MEMORY_POOL_SIZE, int INITIAL_CAPACITY, Pri
```

Error [Checkstyle]: -1
This line is longer than 80 characters. Break it into multiple lines so that it is easier to read.

Error [Checkstyle]: -2
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must b

Error [Checkstyle]: -2
Parameters should be named in camelCase where the name starts with a lowercase letter and the first letter of eac
Rename the parameter 'MEMORY_POOL_SIZE'.

Error [Checkstyle]: -2
Parameters should be named in camelCase where the name starts with a lowercase letter and the first letter of eac
Rename the parameter 'INITIAL_CAPACITY'.

```
        table = new Record[INITIAL_CAPACITY];
        size = 0;
        memoryPoolSize = MEMORY_POOL_SIZE;
        freeBlocks = new int[MEMORY_POOL_SIZE];

        for (int i = 0; i < memoryPoolSize; i++) {
            freeBlocks[i] = -1;
        }

        this.writer = writer;
    }

    public boolean insert(Record record) {
```

☐ **Error [Checkstyle]: -1 (limit exceeded)**
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must b

```
        if (search(record.ID) != null) {
            // Record with the same ID already exists
            return false;
        }

        if (size >= table.length * LOAD_FACTOR_THRESHOLD) {
            expandTable();
        }
        int index = findIndex(record.ID);
        table[index] = record;
        size++;
        return true;
    }

    public Record search(int ID) {
```

☐ **Error [Checkstyle]: 0 (limit exceeded)**
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must b

☐ **Error [Checkstyle]: 0 (limit exceeded)**
Parameters should be named in camelCase where the name starts with a lowercase letter and the first letter of eac
Rename the parameter 'ID'.

```
        int index = findIndex(ID);
        if (table[index] != null && table[index].ID == ID && !table[
```

☐ **Error [Checkstyle]: 0 (limit exceeded)**
This line is longer than 80 characters. Break it into multiple lines so that it is easier to read.

```
            return table[index];
        }
        writer.println("Search FAILED -- There is no record with ID
        return null;
    }

    public boolean delete(int ID) {
```

☐ **Error [Checkstyle]: 0 (limit exceeded)**
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must b

☐ **Error [Checkstyle]: 0 (limit exceeded)**
Parameters should be named in camelCase where the name starts with a lowercase letter and the first letter of eac
Rename the parameter 'ID'.

```
        int index = findIndex(ID);
```

```
            if (table[index] != null && table[index].ID == ID) {
                table[index].deleted = true; // Mark the record as delet
```

Error [Checkstyle]: 0 (limit exceeded)
This line is longer than 80 characters. Break it into multiple lines so that it is easier to read.

```
                size--;
                return true;
            }
            return false;
        }

        private void expandTable() {
            Record[] oldTable = table;
            table = new Record[2 * oldTable.length];
            size = 0;

            writer.println("Hash table expanded to " + table.length + "

        for (Record record : oldTable) {
            if (record != null && !record.deleted) {
                insert(record); // Reinsert non-deleted records
            }
        }
    }

    private int findIndex(int ID) {
```

Error [Checkstyle]: 0 (limit exceeded)
Parameters should be named in camelCase where the name starts with a lowercase letter and the first letter of eac
Rename the parameter 'ID'.

```
        int index = ID % table.length; // 10 % 4
        int step = (((ID / table.length) % (table.length / 2)) * 2)

        while (table[index] != null && table[index].ID != ID) {
            index = (index + step) % table.length; // 4 % 4 = 0, 2 %
            // index = (((ID / table.length) % (table.length / 2)) *
        }
        return index;
    }

    public void printHashTable() {
```

Error [Checkstyle]: 0 (limit exceeded)
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must b

```

        writer.println("HashTable:");
        int count = 0;
        for (int i = 0; i < table.length; i++) {
            Record record = table[i];
            if (record != null) {
                if (record.deleted) {
                    writer.println((i + ": TOMBSTONE"));
                } else {
```

Error [Checkstyle]: 0 (limit exceeded)
This '}' should be alone on a line (i.e. no other code should be after it on the same line).

```
                    writer.println((i + ": " + record.ID));
```

```
                count++;
            }
        }
    }

    writer.println("total records: " + count);

}

public void printMemoryBlocks() {
```

Error [Checkstyle]: 0 (limit exceeded)
The Javadoc for this method or constructor is missing. All visible (i.e. not private) methods and constructors must

```
    // writer.println("FreeBlock List:");

    // for (int block : freeBlocks) {

    // if (block == -1)
```

```
            // continue;
            // else
            // writer.println(block + " ");
            // }

            // writer.println("There are no freeblocks in the memory poo
        }
    }
```