CS 3743 Database
Program 2: Hash File Using Probing for Synonyms

DUE DATE: As shown on Blackboard

Converted to Java by Docrob

This is the second part of a two-part programming assignment. In this part, we are implementing probing to handle synonyms.
This program will use your code from the first programming assignment.
IMPORTANT: make a backup of your entire assignment 1 code base BEFORE starting this assignment. The changes in this assignment may break your assignment 1 functionality.

EXTRA CREDIT OPPORTUNITY: This assignment allows for 10 points of EXTRA CREDIT IF you implement vehicleDelete() (see below) AND you correctly process p2ExtraInput.txt

Files I provide :

- **src/java/main** package
  - **P2Main.java** - contains the main() entry point for the SECOND programming assignment, which invokes your functions. This class will be used I**NSTEAD OF** assignment 1's Main class.
- **src/java/misc package**
  - **MutableInteger.java** - an integer wrapper so that the integer content can be changed inside a called function.
- **p2Input.txt** - stream input file used by the driver to specify what needs to be invoked
- **p2ExtraInput.txt** - used for the extra credit input

This program requires you to make changes to some of your previous functions and add some new ones. All of your work will be done in StudentFunctions.java (and a small modification to pom.xml)

First, change your pom.xml:

change this line:

```
<mainClass>main.Main</mainClass>
```

to:

```
<mainClass>main.P2Main</mainClass>
```

Next, delete your class Main.java from assignment 1 (you made a backup of this project, right?!?)

Next: functions you must code or re-use from assignment 1's StudentFunctions:

```
public static int hashCreate(String fileName, HashHeader hashHeader)
```

Re-use this function. No changes necessary.

```
public static int hashOpen(String fileName, HashFile hashFile)
```

Re-use this function. No changes necessary.

```
public static int readRec(HashFile hashFile, int rbn, Vehicle vehicle)
```

Re-use this function. No changes necessary.

```
public static int writeRec(HashFile hashFile, int rbn, Vehicle vehicle)
```

Re-use this function. No changes necessary.

```
public static int vehicleInsert(HashFile hashFile, Vehicle vehicle)
```

Modify this function starting at the bulleted item in RED

- Determine the RBN using P2Main's hash function.
- Use readRec to read the record at that RBN.
- If that location doesn't exist or the record at that location has a vehicleId[0] == "":
  - Write this new vehicle record at that location using writeRec.
- If that record exists and that vehicle's vehicleId matches, return RC_REC_EXISTS. (Do not update it.)
- Otherwise, it is a synonym to the vehicle in the hashed location:

- Determine if it exists by probing. We use a probing K value of 1. If it does already exist, return RC_REC_EXISTS. (Do not update it.)
- Limit the probing to hashFile.getHashHeader().getMaxProbe(). For example, if …getMaxProbe() returns 3, you can look at the original hash location and at most two additional records. We are only looking at adjacent records below it.
- if there isn't an empty slot and we have probed a total of …getMaxProbe() times (including looking at the hashed location), return RC_TOO_MANY_COLLISIONS. If it doesn't exist and there is an empty slot (maybe because we haven't yet written to that slot), write it to that empty slot.

```
public static int vehicleRead(HashFile hashFile, MutableInteger rbn, Vehicle vehicle)
```

NOTE THE CHANGE IN PARAMETER rbn's data type!!! The MutableInteger class is included in the misc package. Be sure to take a look at it.

Modify this function starting at the bulleted item in RED

- Change your function to use rbn as a MutableInteger
- Since the vehicles vehicleId was provided, determine the RBN using P2Main's hash function.
- Use readRec to read the record at that RBN.
- If the vehicle at that location matches the specified vehicle's vehicleId, return the vehicle via the vehicle parameter and return RC_OK.
- Otherwise, it is a synonym to the vehicle in the hashed location:
  - Determine if it exists as a synonym using probing with a K value of 1.
  - Be sure to store any changed rbn in the rbn parameter!! P2Main uses it.
  - If vehicleIds match, return the vehicle via the vehicle parameter and return RC_OK.
  - If you read past the maximum records in the file, return RC_REC_NOT_FOUND.
  - If you have read for the maximum probes and it wasn't found, return RC_REC_NOT_FOUND.

NEW FUNCTION!!!

```
public static int vehicleUpdate(HashFile hashFile, Vehicle vehicle)
```

This function tries to find the given vehicle using its …getVehicleId(). If found, it updates the contents of the vehicle in the hash file. If not found, it returns RC_REC_NOT_FOUND. Note that this function must understand probing.

NOTE: You can make your life easier with this function if you use MutableInteger and call some of your other functions to help out.

EXTRA CREDIT FUNCTION!!!

```
public static int vehicleDelete(HashFile hashFile, char [] vehicleId)
```

If you did not do the extra credit, create a simple function that just returns `RC_NOT_IMPLEMENTED.`

This function finds the specified vehicle and deletes it by simply setting all bytes in that record to '\0'. Once deleted, this <u>**may**</u> impact your vehicleRead, vehicleInsert, and vehicleUpdate since there can now be empty records along a synonym list even though the needed vehicle could be after it.

Note:

1. Your code must be written in the **StudentFunctions.java** file
2. Do NOT modify ANY of the other files
3. You must run your code on a fox server.
4. TO SUBMIT: zip your src directory and the pom.xml in a single archive file

THIS IS STILL A MAVEN PROJECT

All the previous Maven and JDK info still applies.

To compile your code from the directory above cs3743 (package directory):

```
mvn compile
```

To build (includes compiling) and execute your program:

```
mvn package exec:java
```

SAMPLE OUTPUT

Please see the sample output starting on the next page

GRADING RUBRIC

- (minus) 30% Program 1 bugs cause incorrect output
- (minus) 30% Synonym insertion is incorrect
- (minus) 20% Not properly detecting too many collisions
- (minus) 10% Update is incorrect

EXTRA CREDIT: + (plus) 10 % if vehicles correctly delete

Sample Output (partial):

```
* 8. Insert more vehicles
*
 >> INSERT VEHICLE LH001,LAMBORGHINI,Huracan,2018
            Hash RBN is 9
 >> INSERT VEHICLE SFE09,SUBARU,FE,1979
            Hash RBN is 5
 >> INSERT VEHICLE HA004,HONDA,Accord,1984
            Hash RBN is 1
 >> INSERT VEHICLE FX004,FORD,Expedition,1990
            Hash RBN is 3
 >> INSERT VEHICLE T4004,TOYOTA,4Runner,2004
            Hash RBN is 19
 >> INSERT VEHICLE T4006,TOYOTA,4Runner,2006
            Hash RBN is 2
 >> PRINTALL VEHICLE vehicle.dat
    MaxHash=19, RecSize=36, MaxProbes=5
     1  HA004 1984 HONDA             Accord Hash=1
     2  T4006 2006 TOYOTA           4Runner Hash=2
     3  FX004 1990 FORD          Expedition Hash=3
     5  SFE09 1979 SUBARU                FE Hash=5
     6  TP001 2007 TOYOTA             Prius Hash=6
     9  LH001 2018 LAMBORGHINI      Huracan Hash=9
    13  CB001 1957 CHEVY             BelAir Hash=13
    16  DD001 1970 DODGE               Dart Hash=16
    17  DD002 1974 DODGE               Dart Hash=17
    19  T4004 2004 TOYOTA           4Runner Hash=19
*
* 9. Insert a vehicle that is a synonym
*
 >> INSERT VEHICLE HL001,HONDA,Lift,2013
            Hash RBN is 9
 >> READ VEHICLE HL001
            Hash RBN is 9
    10  HL001 2013 HONDA               Lift Hash=9
 >> READ VEHICLE LH001
            Hash RBN is 9
     9  LH001 2018 LAMBORGHINI      Huracan Hash=9
 >> PRINTALL VEHICLE vehicle.dat
    MaxHash=19, RecSize=36, MaxProbes=5
     1  HA004 1984 HONDA             Accord Hash=1
     2  T4006 2006 TOYOTA           4Runner Hash=2
     3  FX004 1990 FORD          Expedition Hash=3
     5  SFE09 1979 SUBARU                FE Hash=5
     6  TP001 2007 TOYOTA             Prius Hash=6
     9  LH001 2018 LAMBORGHINI      Huracan Hash=9
    10  HL001 2013 HONDA               Lift Hash=9
    13  CB001 1957 CHEVY             BelAir Hash=13
    16  DD001 1970 DODGE               Dart Hash=16
    17  DD002 1974 DODGE               Dart Hash=17
    19  T4004 2004 TOYOTA           4Runner Hash=19
*
* 10. Insert more vehicles that collide
*
 >> INSERT VEHICLE TH008,TOYOTA,Highlander,2015
            Hash RBN is 5
 >> INSERT VEHICLE FF001,FERRARI,F40,1987
            Hash RBN is 1
 >> READ VEHICLE HL001
```

```
           Hash RBN is 9
   10  HL001 2013 HONDA               Lift Hash=9
>> READ VEHICLE TH008
           Hash RBN is 5
    7  TH008 2015 TOYOTA        Highlander Hash=5
```