Class Diagram

STR10

(-)char[10] key

(+)STR10(),STR10(String n),STR10(char[] a),STR10(STR10 n)

(+)get()//return key

(+)set(char[] n)

(+)string()//return string form of char array

STR20

(-)char[20] key

(+)STR20(),STR20(String n),STR20(char[] a),STR20(STR20 n)

(+)get()//return key

(+)set(char[] n)

(+)string()//return string form of char array

SLOT

(-)STR10 KEY, STR20 DATA

(+)SLOT(), SLOT(STR10 a, STR20 b), SLOT(char[] a, char[] b)

(+)getDATA()

(+)getKEY()

(+)setDATA()

(+)setKEY()

BUCKET

(-)short COUNTER, OVERFLOW

(-)SLOT[30] container

(+)BUCKET(), BUCKET(SLOT[] n)

(+)OVERFLOW()//adds counter to overflow

(+)clear()//resets counter and overflow counter

(+)getDATA(short slotIndex)

(+)getKEY(short slotIndex)

(+)getOVERFLOW()

(+)getSLOT()

(+)insertSLOT()

(+)isFULL()

(+)setDATA(STR20 d, short slotIndex))

(+)setKEY(STR10 k, short slotIndex))

(+)setSLOT(STR10 key, STR20 data, short slotIndex)

BUCKET—Contains(3) 🡪 SLOT

HashTable

(-)BUCKETS[] container

(-) short overflowBUCKETS, overflowINCREMENTS, primaryBUCKETS

(+)GenerateStatReport(String name)

(+)HashFunction(STR10 key, short tablesize)

(+)RestoreHTtoMem(File file)

(+)SearchHT(File file)

(+)WriteHTtoDisk()

(+)avgOverflow()

(+)initialize()

(+)insertIntoHT(STR10 k, STR20 d)

(+)returnBucket(STR10 k, int position)

(+)search(STR10 k)

(+)searchBucket(STR10 k, int position)

HashTable –Contains(1) 🡪 BUCKETS[30]

AppDriver – Contains(1) 🡪 HashTable

AppDriver – Contains(3) 🡪 File

Abnormal Use-Case Scenario 1

1. Parse DATA file
2. DATA file is corrupt
3. Ask user to verify SEARCH file.
4. Search File is correct, parse HT – Continue for main Code

Abnormal Use-Case Scenario 2

1. Parse SEARCH file
2. Write to hash table
3. Generate Stat Report
4. Write to disk
5. Restore from disk
6. Search HT
   1. Corrupt search file
   2. Ask user to verify search file