LinkedListMiddleware calling methods:

Overloaded constructor:

LinkedListMiddleware(int listSize)

Methods:

selectSort()

sorts, then prints out time taken

bubbleSort()

sorts, then prints out time taken

linearSearch(int value)

searches, displays IF value is found, then time taken to search

binarySearch(int value)

searches, displays WHERE value is found, then time taken to search, then iterations through list

method names for Part 2:

public StackCut(int param)

public Object push(Object param)

public Object pop()

public void displayStacks()

public Object peek()

public int getID()

public String getName()

public String getMajor()

public double getGPA()

public int getTG()

public void setUp()

public void testStudent()

public void testStack()

Constructor for student class:

public Student(int gID, String name, String major, double gpa, int tg)

LinkListDatabase calling methods:

Constructor

LinkedList()

Methods:

boolean isEmpty()

void insertFirstLink(T obj)

Link removeFirst()

removes first link

Link find(T obj)

finds link object and returns it

Link removeLink(T obj)

removes specified object from LinkedList

Link get(int id)

returns Link object at specified id

void setLink(int index, int value)

replaces previous value at the index with given value

**Public Methods:**

* public void createBinaryTree(int size)
* public String calculatePreorderNodesSearched(int key1, int key2, int key3)
* public String calculateInOrderNodesSearched(int key1, int key2, int key3)
* public String calculatePreorderSearchDuration(int key1, int key2, int key3)
* public String calculateInOrderSearchDuration(int key1, int key2, int key3)
* public String calculateBigORelationship(BinaryTree binaryTree)
* public Database getDatabase()
* public void setDatabase(Database database)

**HashClient methods:**int createHashTable(int n);

void findHashTableValue();  
  
String bigONotation();

int findHashTableValue();

**HashFinder methods:**

int findHashValue(int input, int size, int array);

MidArrayList methods:

MidArrayList(int i); overloaded constructor

generateRandomArray(); This generates a random array of students.

returnArraySize(); Returns the size of the array of students

bubbleSort(); Bubble sorts the array into ascending order. Prints the time taken to sort in ms.

clearArray(); Clears the array of students.

selectionSort(); Selection sorts the array into ascending order. Prints the time taken to sort in ms.

binarySearch(int i); Binary searches for a value in a sorted array. This takes an integer (the value to search for)

linearSearch(int i); linear searches for a value in a sorted array. This takes an integer (the value to search for)

returnArray(); returns the array in its entirety. Used for debugging.

**HashClient methods:**int createHashTable(int n);

void findHashTableValue();  
  
String bigONotation();

**HashFinder methods:**

int findHashValue(int input, int size, int array);