2BAirNot2B

Version 1
April 12, 2020

Group 4
Michael Yohannes
Aly Shah Imtiaz
Bryan Chiu

Comp Sci 2XB3 - L01
Department of Computing and Software - McMaster University

Chapter 1

Revision Page

Revision History: None

Team Members:
Micheal Yohannes: 001316842
Aly Shah Imtiaz: 400030195
Bryan Chiu: 400036252

Roles and Responsibilities:

Micheal: Was responsible for the majority of the code. Was responsible for ideation and finding datasets to use.

Aly Shah: Was responsible for half of the design specifications document, and half of the requirements specifications document. Was also responsible for design of the system and classes to be used.

Bryan: Was responsible for half of the design specifications document, and half of the requirements specifications document. Was also responsible for ensuring all members were on track for their deliverables.

By virtue of submitting this document we electronically sign and date that the work being submitted by all the individuals in the group is their exclusive work as a group and we consent to make available the application developed through CS-2XB3 project, the reports, presentations, and assignments (not including my name and student number) for future teaching purposes.

Chapter 2

Contribution Page

Name	Role	Contributions	Comments
Michael Yohannes	Software Architect	• Implemented the algorithms (Graph: BST; Sorting: Quick Sort; Searching: Binary Search). • Implemented the User Interface. • Implemented the CSV scraping and parsing.	
Aly Shah Imtiaz	Project Manager	• Created the general architecture of the project. • Added JavaDocs comments to the codebase. • Worked on the requirements and design specifications document. • Responsible for the overall design of the system and classes to be used.	
Bryan Chiu	Team Lead	• Created the base model for the User Interface and outlined how the UI would be shown in the terminal. • Added JavaDocs comments to the codebase. • Worked on the requirements and design specifications document. • Ensured deadlines were met.	

Chapter 3

Executive Summary

With the significant growth in renters and decrease in homeowners in the past decade, it is now more profitable than ever to rent out housing in metropolitan areas. Despite the high demand, different factors affect the profitability of a property, and entrepreneurs looking to purchase a property to rent out need to know these factors. It is also important for landlords/landladies to understand what makes a listing popular and optimize their existing properties to attract more potential renters. 2BAirNot2B uses AirBnB listings data to provide users with property information that they can use to make purchases and increase their profits. 2BAirNot2B uses deterministic algorithms to extract data from open datasets of AirBnB listings of major metropolitan areas. The user will query for listings with specified filters, and 2BAirNot2B will efficiently return listings that meet the query criteria. Using 2BAirNot2B, entrepreneurs will gain the insight to prosper in the real estate market.

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 5

Class Index

5.1 Class List

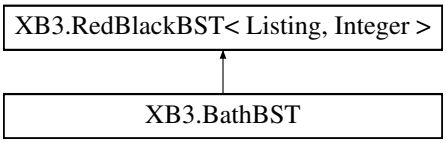
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 6

Class Documentation

6.1 XB3.BathBST Class Reference

Inheritance diagram for XB3.BathBST:



Public Member Functions

- Queue< Listing > getRangeToDelete (Filter someFilter)
- Queue< Listing > keys (Double lo, Double hi)

Static Public Attributes

- static final String **treeType** = "bathrooms"

Additional Inherited Members

6.1.1 Detailed Description

BST for listings based on number of bathrooms

Author

Michael Yohannes

6.1.2 Member Function Documentation

6.1.2.1 getRangeToDelete()

```
Queue<Listing> XB3.BathBST.getRangeToDelete (
            Filter someFilter )  [inline]
```

Returns all keys in the RedBlackBST before/after a certain cutoff, as a Queue

Parameters

someFilter	filter

Returns

either all keys before/after `someFilter`

(inclusive) as a `Queue`

, or `null`

6.1.2.2 keys()

```
Queue<Listing> XB3.BathBST.keys (
            Double lo,
            Double hi )  [inline]
```

Returns all keys in the RedBlackBST in the given range, as a Queue

Parameters

lo	minimum endpoint
hi	maximum endpoint

Returns

all keys in the symbol table between `lo`

(inclusive) and `hi`

(inclusive) as a `Queue`

Exceptions

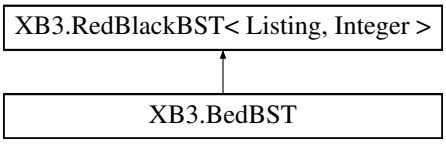
IllegalArgumentException	if either `lo` or `hi` is `null`

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/BathBST.java

6.2 XB3.BedBST Class Reference

Inheritance diagram for XB3.BedBST:



Public Member Functions

- Queue< Listing > getRangeToDelete (Filter someFilter)
- Queue< Listing > keys (Double lo, Double hi)

Static Public Attributes

- static final String **treeType** = "bedrooms"

Additional Inherited Members

6.2.1 Detailed Description

BST for listings based on number of bedrooms

Author

Michael Yohannes

6.2.2 Member Function Documentation

6.2.2.1 getRangeToDelete()

```
Queue<Listing> XB3.BedBST.getRangeToDelete (
            Filter someFilter )  [inline]
```

Returns all keys in the RedBlackBST before/after a certain cutoff, as a Queue

Parameters

someFilter	filter

Returns

either all keys before/after
`someFilter`

(inclusive) as a
`Queue`

, or
`null`

6.2.2.2 keys()

```
Queue<Listing> XB3.BedBST.keys (
            Double lo,
            Double hi )  [inline]
```

Returns all keys in the RedBlackBST in the given range, as a Queue

Parameters

lo	minimum endpoint
hi	maximum endpoint

Returns

all keys in the symbol table between
`lo`
(inclusive) and
`hi`
(inclusive) as a
`Queue`

Exceptions

IllegalArgumentException	if either `lo` or `hi` is `null`

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/BedBST.java

6.3 XB3.Filter Class Reference

Public Member Functions

- String getQuery ()
- String getRelation ()
- String getValue ()
- boolean **equals** (Object obj)
- int **hashCode** ()

Static Public Member Functions

- static boolean validFilter (String str)

6.3.1 Detailed Description

Class that represents properties used to filter listings

Author

Michael Yohannes

6.3.2 Member Function Documentation

6.3.2.1 getQuery()

```
String XB3.Filter.getQuery ( )  [inline]
```

Returns the property to be filtered, as a String

Returns

the property to be filtered, as a String

6.3.2.2 getRelation()

```
String XB3.Filter.getRelation ( )  [inline]
```

Returns the relation to the property to be filtered, either "<" or ">"

Returns

the relation to the property to be filtered, either "<" or ">"

6.3.2.3 getValue()

```
String XB3.Filter.getValue ( )  [inline]
```

Returns the value of the property to be filtered, as a String

Returns

the value of the property to be filtered, as a String

6.3.2.4 validFilter()

```
static boolean XB3.Filter.validFilter (
            String str )  [inline], [static]
```

Is the Filter valid?

Parameters

str	input String

Returns

whether the input String can be parsed into a valid Filter

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/Filter.java

6.4 XB3.FilterLive Class Reference

Public Member Functions

- HashMap< String, RedBlackBST< Listing, Integer > > getLiveTrees ()
- void addFilter (String someFilter) throws Exception
- void deleteListings (ArrayList< Listing > rangeToDelete)
- void removeFilter (String someFilter) throws Exception
- void addAnFFile (Integer seqInteger, Integer FfileInteger) throws FileNotFoundException, CsvValidation↩ Exception, IOException
- boolean liveTreeInvariant ()
- void showTop (int x)
- String toString ()

6.4.1 Detailed Description

Class that represents properties used to filter listings

Author

Michael Yohannes

6.4.2 Member Function Documentation

6.4.2.1 addAnFFile()

```
void XB3.FilterLive.addAnFFile (
            Integer seqInteger,
            Integer FfileInteger ) throws FileNotFoundException, CsvValidationException, IO↩
Exception  [inline]
```

Determines the indices to delete and adds it to the set and files accordingly.

Parameters

seqInteger	sequence index
FfileInteger	Ffile index

Exceptions

FileNotFoundException	if file is not found
CsvValidationException	if CSV is not validated appropriately
IOException	if I/O exception occurs

6.4.2.2 addFilter()

```
void XB3.FilterLive.addFilter (
            String someFilter ) throws Exception  [inline]
```

Adds a filter to the tree selected.

Parameters

someFilter	filter to be added

Exceptions

Exception	if trees are not balanced

6.4.2.3 deleteListings()

```
void XB3.FilterLive.deleteListings (
            ArrayList< Listing > rangeToDelete )  [inline]
```

Deletes listings in the given tree.

Parameters

rangeToDelete	listings to be deleted

6.4.2.4 getLiveTrees()

```
HashMap<String, RedBlackBST<Listing, Integer> > XB3.FilterLive.getLiveTrees ( )  [inline]
```

Returns a HashMap of String and a BST.

Returns

a HashMap of String and a BST.

6.4.2.5 liveTreeInvariant()

```
boolean XB3.FilterLive.liveTreeInvariant ( )  [inline]
```

Are all BSTs the same size?

Returns

whether the BSTs for each property are the same.

6.4.2.6 removeFilter()

```
void XB3.FilterLive.removeFilter (
            String someFilter ) throws Exception  [inline]
```

Removes a filter from the current query.

Parameters

someFilter	filter to be removed

Exceptions

Exception	if addAnFFile fails

6.4.2.7 showTop()

```
void XB3.FilterLive.showTop (
            int x )  [inline]
```

Prints out each listing in the BST by revenue.

6.4.2.8 toString()

```
String XB3.FilterLive.toString ( )  [inline]
```

Returns a String representation of the number of listings matching the query.

Returns

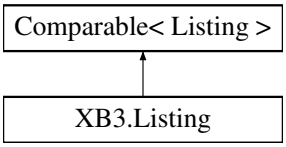
a String representation of the number of listings matching the query.

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/FilterLive.java

6.5 XB3.Listing Class Reference

Inheritance diagram for XB3.Listing:



Public Member Functions

- Double validNumber (String val)
- String[ ] toSeq ()
- int getId ()
- boolean **isSuperHost** ()
- String **getNeighborhood** ()
- String **getZipcode** ()
- String **getPropType** ()
- Double **getAccommodates** ()
- Double **getBathrooms** ()
- Double **getBedrooms** ()
- Double **getDayPrice** ()
- Double **getReviewRating** ()
- Double **getAvail365** ()
- BigDecimal **getRevenue** ()
- Map< String, String > **getListingData** ()
- String **toString** ()
- int compareTo (Listing obj)
- int **compareRev** (Listing that)
- int **compareBed** (Listing that)
- int **compareBath** (Listing that)

Static Public Member Functions

- static Listing changeTreeType (Listing listing, String treeType)

6.5.1 Detailed Description

Class that represents an AirBNB listing

Author

Michael Yohannes

6.5.2 Member Function Documentation

6.5.2.1 changeTreeType()

```
static Listing XB3.Listing.changeTreeType (
            Listing listing,
            String treeType )  [inline], [static]
```

Takes a listing and returns it with a changed tree type.

Parameters

listingData	listing data from csv fields
treeType	tree type

6.5.2.2 compareTo()

```
int XB3.Listing.compareTo (
            Listing obj )  [inline]
```

compareTo methods for revenue, bedrooms, bathrooms

6.5.2.3 getId()

```
int XB3.Listing.getId ( )  [inline]
```

Get methods for the listing data

6.5.2.4 toSeq()

```
String [] XB3.Listing.toSeq ( )  [inline]
```

Returns the listing data as an array of String.

Returns

the listing data as an array of String

6.5.2.5 validNumber()

```
Double XB3.Listing.validNumber (
            String val )  [inline]
```

Attempts to parse an input value into a double.

Parameters

val	String value to be parsed

Returns

String value parsed into a double

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/Listing.java

6.6 XB3.Main Class Reference

Static Public Member Functions

- static void **main** (String[ ] args) throws Exception

6.6.1 Detailed Description

Main class for 2BAirNot2B Run and follow prompts

Author

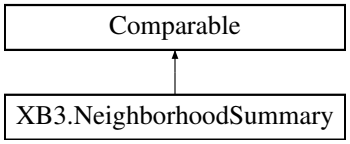
Michael Yohannes

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/Main.java

6.7 XB3.NeighborhoodSummary Class Reference

Inheritance diagram for XB3.NeighborhoodSummary:



Public Member Functions

- int **compareTo** (Object obj)
- String **getNeighborhood** ()
- DescriptiveStatistics **getNeighborhoodData** ()

6.7.1 Detailed Description

Class that represents a neighbourhood

Author

Michael Yohannes

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/NeighborhoodSummary.java

6.8 XB3.RedBlackBST< Key extends Comparable< Key, Value >.Node Class Reference

Public Member Functions

- **Node** (Key key, Value val, boolean color, int size)

Protected Attributes

- Key **key**
- Value **val**
- Node **left**
- boolean **color**
- int **size**

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/RedBlackBST.java

6.9 XB3.Optimization Class Reference

Public Member Functions

- Optimization (int id) throws FileNotFoundException, CsvValidationException, IOException, InvalidValue
- void showTopZipCodes ()
- void showTopNeighborhoods ()
- void optimizeReport ()

6.9.1 Detailed Description

The optimization feature of 2BAirNot2B

Author

Michael Yohannes

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Optimization()

```
XB3.Optimization.Optimization (
            int id ) throws FileNotFoundException, CsvValidationException, IOException, Invalid↩
Value  [inline]
```

Parameters

id	the id

Exceptions

FileNotFoundException	Raises an error if the File is unable to be found.
CsvValidationException	Raises an error if the CSV is unable to be validated appropriately.
IOException	Raises an error if an input/output exception occurs while writing to the file.
InvaldValue	Raises an error if the value is not valid.

6.9.3 Member Function Documentation

6.9.3.1 optimizeReport()

```
void XB3.Optimization.optimizeReport ( )  [inline]
```

A way to optimize the report and get data faster for larger datasets.

6.9.3.2 showTopNeighborhoods()

```
void XB3.Optimization.showTopNeighborhoods ( )  [inline]
```

Shows neighborhoods better than current ID.

6.9.3.3 showTopZipCodes()

```
void XB3.Optimization.showTopZipCodes ( )  [inline]
```

Shows zipcodes better than current ID.

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/Optimization.java

6.10 XB3.ReadListingsFromCSV Class Reference

Static Public Member Functions

- static ArrayList< Listing > ReadListings (String file) throws FileNotFoundException, IOException, Csv↩ ValidationException

6.10.1 Detailed Description

Reads AirBnB listings from CSV

Author

Michael Yohannes

6.10.2 Member Function Documentation

6.10.2.1 ReadListings()

```
static ArrayList<Listing> XB3.ReadListingsFromCSV.ReadListings (
            String file ) throws FileNotFoundException, IOException, CsvValidationException
[inline], [static]
```

Opens the CSV file and creates an ArrayList of Listing objects, and returns the array.

Parameters

file	is a String corresponding to the filename.

Returns

Returns an ArrayList of Listing objects.

Exceptions

FileNotFoundException	Raises an error if the File is unable to be found.
IOException	Raises an error if an input/output exception occurs while writing to the file.
CsvValidationException	Raises an error if the CSV is unable to be validated appropriately.

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/ReadListingsFromCSV.java

6.11 XB3.RedBlackBST< Key extends Comparable< Key, Value > Class Template Reference

Classes

- class Node

Public Member Functions

- RedBlackBST ()
- int size ()
- boolean isEmpty ()
- Value get (Key key)
- boolean contains (Key key)
- void put (Key key, Value val)
- void deleteMin ()
- void deleteMax ()
- void delete (Key key)
- int height ()
- Key min ()
- Key max ()
- Key floor (Key key)
- Key ceiling (Key key)
- Key select (int rank)
- int rank (Key key)
- Iterable< Key > keys ()
- Iterable< Key > keys (Key lo, Key hi)
- int size (Key lo, Key hi)
- Queue< Key > **inorder** ()
- Queue< Key > **revInorder** (int top)

Protected Attributes

- Node **root**

Static Protected Attributes

- static final boolean **RED** = true
- static final boolean **BLACK** = false

6.11.1 Detailed Description

Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne.

Author

Robert Sedgewick

Kevin Wayne

6.11.2 Constructor & Destructor Documentation

6.11.2.1 RedBlackBST()

```
XB3.RedBlackBST< Key extends Comparable< Key, Value >.RedBlackBST ( )  [inline]
```

Initializes an empty symbol table.

6.11.3 Member Function Documentation

6.11.3.1 ceiling()

```
Key XB3.RedBlackBST< Key extends Comparable< Key, Value >.ceiling (
            Key key )  [inline]
```

Returns the smallest key in the symbol table greater than or equal to
`key`

.

Parameters

key	the key

Returns

the smallest key in the symbol table greater than or equal to
`key`

Exceptions

NoSuchElementException	if there is no such key
IllegalArgumentException	if `key` is `null`

6.11.3.2 contains()

```
boolean XB3.RedBlackBST< Key extends Comparable< Key, Value >.contains (
            Key key )  [inline]
```

Does this symbol table contain the given key?

Parameters

key	the key

Returns

`true`

if this symbol table contains `key`

and `false`

otherwise

Exceptions

IllegalArgumentException	if `key` is `null`

6.11.3.3 delete()

```
void XB3.RedBlackBST< Key extends Comparable< Key, Value >.delete (
            Key key )  [inline]
```

Removes the specified key and its associated value from this symbol table (if the key is in this symbol table).

Parameters

key	the key

Exceptions

IllegalArgumentException	if `key` is `null`

6.11.3.4 deleteMax()

```
void XB3.RedBlackBST< Key extends Comparable< Key, Value >.deleteMax ( )  [inline]
```

Removes the largest key and associated value from the symbol table.

Exceptions

NoSuchElementException	if the symbol table is empty

6.11.3.5 deleteMin()

```
void XB3.RedBlackBST< Key extends Comparable< Key, Value >.deleteMin ( )  [inline]
```

Red-black tree deletion. Removes the smallest key and associated value from the symbol table.

Exceptions

NoSuchElementException	if the symbol table is empty

6.11.3.6 floor()

```
Key XB3.RedBlackBST< Key extends Comparable< Key, Value >.floor (
            Key key )  [inline]
```

Returns the largest key in the symbol table less than or equal to
`key`

.

Parameters

key	the key

Returns

the largest key in the symbol table less than or equal to
`key`

Exceptions

NoSuchElementException	if there is no such key
IllegalArgumentException	if `key` is `null`

6.11.3.7 get()

```
Value XB3.RedBlackBST< Key extends Comparable< Key, Value >.get (
            Key key )  [inline]
```

Standard BST search. Returns the value associated with the given key.

Parameters

key	the key

Returns

the value associated with the given key if the key is in the symbol table and `null`

if the key is not in the symbol table

Exceptions

IllegalArgumentException	if `key` is `null`

6.11.3.8 height()

```
int XB3.RedBlackBST< Key extends Comparable< Key, Value >.height ( )  [inline]
```

Utility functions. Returns the height of the BST (for debugging).

Returns

the height of the BST (a 1-node tree has height 0)

6.11.3.9 isEmpty()

```
boolean XB3.RedBlackBST< Key extends Comparable< Key, Value >.isEmpty ( )  [inline]
```

Is this symbol table empty?

Returns

`true`

if this symbol table is empty and `false`

otherwise

6.11.3.10 keys() [1/2]

```
Iterable<Key> XB3.RedBlackBST< Key extends Comparable< Key, Value >.keys ( )  [inline]
```

Range count and range search. Returns all keys in the symbol table as an `Iterable`

. To iterate over all of the keys in the symbol table named `st`

, use the foreach notation:

```
for (Key key : st.keys())
```

.

Returns

all keys in the symbol table as an `Iterable`

6.11.3.11 keys() [2/2]

```
Iterable<Key> XB3.RedBlackBST< Key extends Comparable< Key, Value >.keys (
            Key lo,
            Key hi )  [inline]
```

Returns all keys in the symbol table in the given range, as an
`Iterable`

.

Parameters

lo	minimum endpoint
hi	maximum endpoint

Returns

all keys in the symbol table between
`lo`

(inclusive) and
`hi`

(inclusive) as an
`Iterable`

Exceptions

IllegalArgumentException	if either `lo` or `hi` is `null`

6.11.3.12 max()

```
Key XB3.RedBlackBST< Key extends Comparable< Key, Value >.max ( )  [inline]
```

Returns the largest key in the symbol table.

Returns

the largest key in the symbol table

Exceptions

NoSuchElementException	if the symbol table is empty

6.11.3.13 min()

```
Key XB3.RedBlackBST< Key extends Comparable< Key, Value >.min ( )  [inline]
```

Ordered symbol table methods. Returns the smallest key in the symbol table.

Returns

the smallest key in the symbol table

Exceptions

NoSuchElementException	if the symbol table is empty

6.11.3.14 put()

```
void XB3.RedBlackBST< Key extends Comparable< Key, Value >.put (
            Key key,
            Value val )  [inline]
```

Red-black tree insertion. Inserts the specified key-value pair into the symbol table, overwriting the old value with the new value if the symbol table already contains the specified key. Deletes the specified key (and its associated value) from this symbol table if the specified value is
`null`

.

Parameters

key	the key
val	the value

Exceptions

IllegalArgumentException	if `key` is `null`

6.11.3.15 rank()

```
int XB3.RedBlackBST< Key extends Comparable< Key, Value >.rank (
            Key key )  [inline]
```

Return the number of keys in the symbol table strictly less than
`key`

.

Parameters

key	the key

Returns

the number of keys in the symbol table strictly less than
`key`

Exceptions

IllegalArgumentException	if `key` is `null`

6.11.3.16 select()

```
Key XB3.RedBlackBST< Key extends Comparable< Key, Value >.select (
            int rank )  [inline]
```

Return the key in the symbol table of a given
`rank`

. This key has the property that there are
`rank`

keys in the symbol table that are smaller. In other words, this key is the (
`rank`

+1)st smallest key in the symbol table.

Parameters

rank	the order statistic

Returns

the key in the symbol table of given
`rank`

Exceptions

IllegalArgumentException	unless `rank` is between 0 and n-1

6.11.3.17 size() [1/2]

```
int XB3.RedBlackBST< Key extends Comparable< Key, Value >.size ( )  [inline]
```

Returns the number of key-value pairs in this symbol table.

Returns

the number of key-value pairs in this symbol table

6.11.3.18 size() [2/2]

```
int XB3.RedBlackBST< Key extends Comparable< Key, Value >.size (
            Key lo,
            Key hi )  [inline]
```

Returns the number of keys in the symbol table in the given range.

Parameters

lo	minimum endpoint
hi	maximum endpoint

Returns

the number of keys in the symbol table between
`lo`
(inclusive) and
`hi`
(inclusive)

Exceptions

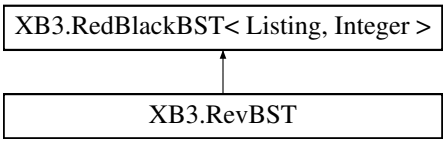
IllegalArgumentException	if either `lo` or `hi` is `null`

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/RedBlackBST.java

6.12 XB3.RevBST Class Reference

Inheritance diagram for XB3.RevBST:



Public Member Functions

- Queue< Listing > getRangeToDelete (Filter someFilter)
- Queue< Listing > keys (BigDecimal lo, BigDecimal hi)

Static Public Attributes

- static final String **treeType** = "revenue"

Additional Inherited Members

6.12.1 Detailed Description

BST for listings based on revenue

Author

Michael Yohannes

6.12.2 Member Function Documentation

6.12.2.1 getRangeToDelete()

```
Queue<Listing> XB3.RevBST.getRangeToDelete (
            Filter someFilter )  [inline]
```

Returns all keys in the RedBlackBST before/after a certain cutoff, as a Queue

Parameters

someFilter	filter

Returns

either all keys before/after
`someFilter`

(inclusive) as a
`Queue`

, or
`null`

6.12.2.2 keys()

```
Queue<Listing> XB3.RevBST.keys (
            BigDecimal lo,
            BigDecimal hi )  [inline]
```

Returns all keys in the RedBlackBST in the given range, as a Queue

Parameters

lo	minimum endpoint
hi	maximum endpoint

Returns

all keys in the symbol table between
`lo`
(inclusive) and
`hi`
(inclusive) as a
`Queue`

Exceptions

IllegalArgumentException	if either `lo` or `hi` is `null`

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/RevBST.java

6.13 XB3.SummaryStats Class Reference

Public Member Functions

- void showNeighborhoods ()
- void showZipCodes ()

6.13.1 Detailed Description

Class that represents a summary of data from appropriate listings

Author

Michael Yohannes

6.13.2 Member Function Documentation

6.13.2.1 showNeighborhoods()

```
void XB3.SummaryStats.showNeighborhoods ( )  [inline]
```

Shows all of the neighborhood data and the mean and standard deviation data.

6.13.2.2 showZipCodes()

```
void XB3.SummaryStats.showZipCodes ( )  [inline]
```

Shows all of the zipcode data and the mean and standard deviation data.

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/SummaryStats.java

6.14 XB3.WriteListingsToCSV Class Reference

Static Public Member Functions

- static void WriteListings (ArrayList< Listing > listings, String file) throws IOException

6.14.1 Detailed Description

Writes all the listings to an output CSV file

Author

Michael Yohannes

6.14.2 Member Function Documentation

6.14.2.1 WriteListings()

```
static void XB3.WriteListingsToCSV.WriteListings (
            ArrayList< Listing > listings,
            String file ) throws IOException  [inline], [static]
```

Writes all the listings to an output CSV file, based on the listings provided in the input parameter.

Parameters

file	name of output file
listings	listings corresponding to cleaned data from ZipCodeClean class

Exceptions

IOException	if the file cannot be created, or other similar file issues

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/WriteListingsToCSV.java

6.15 XB3.ZipCodeClean Class Reference

Static Public Member Functions

- static void writeCleanedListings (ArrayList< Listing > dirtyData) throws IOException
- static HashMap< String, LinkedList< Listing > > uncleanedData (ArrayList< Listing > dirtyData)
- static ArrayList< Listing > cleanData (HashMap< String, LinkedList< Listing >> zipcodes)
- static boolean validFields (Listing listing)

6.15.1 Detailed Description

Cleans listings based on zipcode and writes to new CSV

Author

Michael Yohannes

6.15.2 Member Function Documentation

6.15.2.1 cleanData()

```
static ArrayList<Listing> XB3.ZipCodeClean.cleanData (
            HashMap< String, LinkedList< Listing >> zipcodes )  [inline], [static]
```

Returns an ArrayList of Listings after processing the mean and standard deviation.

Parameters

zipcodes	hashmap of string to listing linked list, corresponding to zip codes

Returns

an ArrayList of Listings after processing the mean and standard deviation

6.15.2.2 uncleanedData()

```
static HashMap<String, LinkedList<Listing> > XB3.ZipCodeClean.uncleanedData (
            ArrayList< Listing > dirtyData )  [inline], [static]
```

Returns a HashMap with the key being the zipcode and a linked list of listings which correspond to that zip code.

Parameters

dirtyData	an ArrayList of Listings with uncleaned data.

Returns

a HashMap with the key being the zipcode and a linked list of listings which correspond to that zip code

6.15.2.3 validFields()

```
static boolean XB3.ZipCodeClean.validFields (
            Listing listing )  [inline], [static]
```

Determines whether all the fields are valid in a listing.

Parameters

listing	listing to check

Returns

whether all the fields are valid in a listing

6.15.2.4 writeCleanedListings()

```
static void XB3.ZipCodeClean.writeCleanedListings (
            ArrayList< Listing > dirtyData ) throws IOException  [inline], [static]
```

Writes the listing to the output CSV.

Parameters

dirtyData	uncleaned data

Exceptions

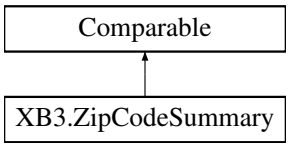
IOException	if the CSV file can not be opened

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/ZipCodeClean.java

6.16 XB3.ZipCodeSummary Class Reference

Inheritance diagram for XB3.ZipCodeSummary:



Public Member Functions

- int **compareTo** (Object obj)
- String **getZipcode** ()
- DescriptiveStatistics **getZipcodeData** ()

6.16.1 Detailed Description

Class that represents a zipcode

Author

Michael Yohannes

The documentation for this class was generated from the following file:

- 2XB3_project/src/XB3/ZipCodeSummary.java

Index