CSE222

HW6 Report

Muhammed Eyüp

1801042679

Q1

Class diagram:

```
                    ┌──────────────┐
          ┌────────▶│  BinaryTree  │◀────────┐
          │         └──────────────┘         │
          │                ▲                  │
          │                │                  │
┌─────────────┐    ┌──────────────┐    ┌──────────────┐
│ RedBlackTree │    │   AVLTree    │    │   2-3Tree    │
└─────────────┘    └──────────────┘    └──────────────┘
```
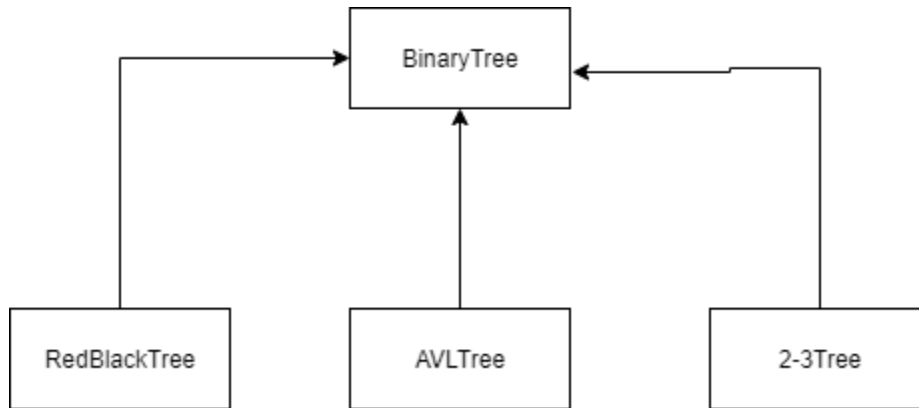
Problem solution approach:

The problem was to insert a set of integers to a set of data structures and the were AVL Tree, Red Black Tree, B tree, 2-3 tree and a skip list, the set of integers were inserted according to the rules of each data structure and removed in FIFO format
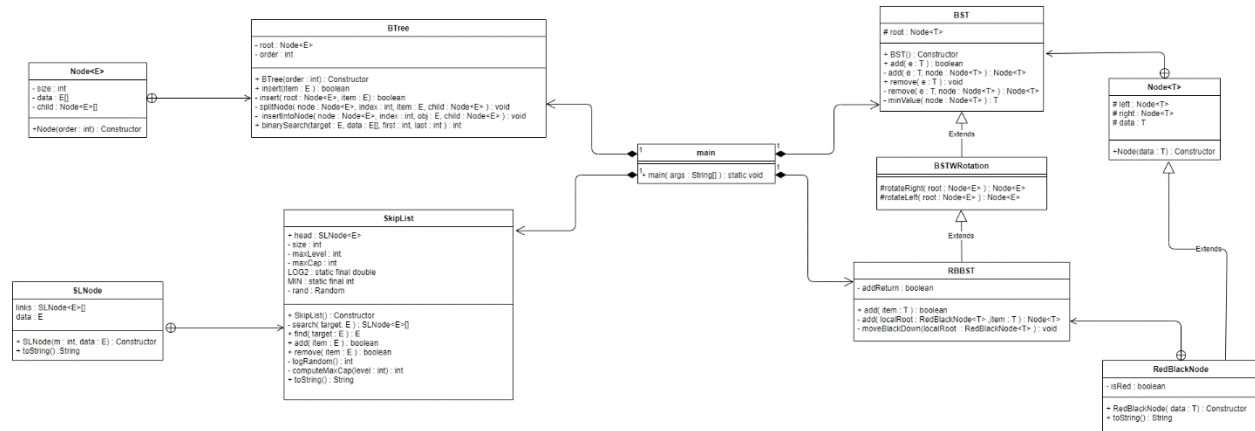
Test Cases:

There were no test cases as there wasn't any code written

Running commands and result:

There were no running commands and results as there wasn't any code written

Q3

Class Diagram:

Problem solution approach:

The problem was to test the performance of the self balancing tree data structures and the binary search tree. each data structure had an object with a size of 10000,20000,40000 and 80000 then for each data structure 10 more integers were inserted and the time it took to insert them was printed as an output then the 10 integers were deleted and the time it took to delete them was also printed as an output

Test cases:

Test scenario	Test case	Result
Insert an element	Element provided	Element inserted
Remove an element	Element provided	Element removed

Running command and results:

```
testing insertion time for each data structure...
BST10k: 0.0082 ms
BST20k: 0.0091 ms
BST40k: 0.0116 ms
BST80k: 0.0116 ms
avg insertion time for BST: 0.010125 ms


RBBST10k: 0.0078 ms
RBBST20k: 0.0117 ms
RBBST40k: 0.0132 ms
RBBST80k: 0.0181 ms
avg insertion time for RBBST: 0.0127 ms


javaRBBST10k: 0.0058 ms
javaRBBST20k: 0.0064 ms
javaRBBST40k: 0.0092 ms
javaRBBST80k: 0.0102 ms
avg insertion time for javaRBBST: 0.0079 ms


SL10k: 0.0086 ms
SL20k: 0.0113 ms
SL40k: 0.0114 ms
SL80k: 0.0243 ms
avg insertion time for javaRBBST: 0.0139 ms
```
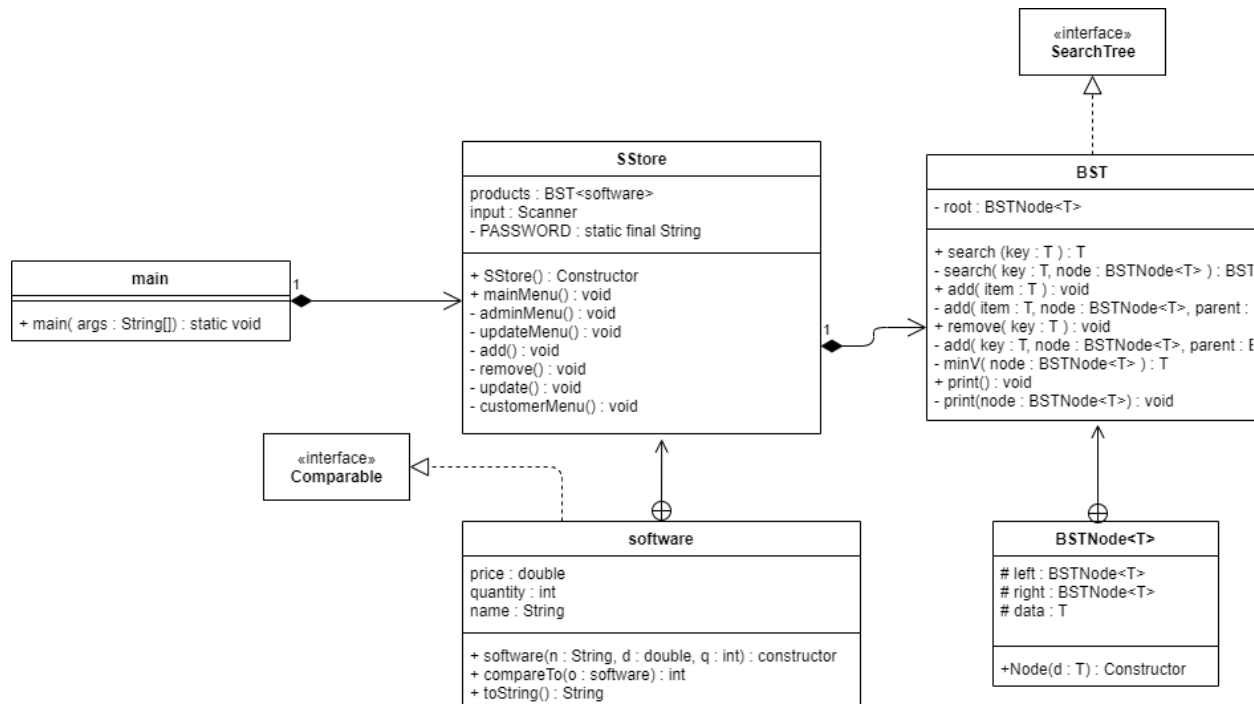
Q4

Class Diagram:

Problem solution approach:

The problem was to create a menu driven software store. The store was created using a binary search tree to store the products but it has an interface to implement any type of tree the user wants.

The program has two types of login as an administrator or as a customer.

The admin needs a password to login and can modify the stores products.

The customer can only buy products.

Test cases:

Test scenario	Test case	Result
Login as an admin	Correct password	Login success
	Wrong password	Gives a warning and asks again
Add a product	Product parameters provided	Adds the product
Remove a product	Product name provided	Removes the product
Updating a product	Product name provided	Asks the admin what to update
Buy a product	Product name provided	Buys the product and decrease quantity by one

Running commands and results:

```
Choose Login type:
0- exit
1- Admin
2- Customre
Choice: 1

Admin Login Menu

please enter a password: 12345
choose what you want to do:
0- up
1- add a product
2- remove a product
 3- update a product
choice: 1
enter the name of the product: avira
enter the price of the product: 49.99
enter the quantity of the product: 1
```

```
choose what you want to do:
0- up
1- add a product
2- remove a product
 3- update a product
choice: 3
Adobe Photoshop 6.0 29.99 1

Adobe Flash 3.3 9.99 1

Adobe Photoshop 6.2 39.99 1

Norton 4.5 19.99 1

Norton 5.5 29.99 1

avira 49.99 1

enter the name of the product: avira
choose what to do:
0-up
1-change name
2-change price
3-change quantity
Choice: 3
enter new quantity: 5
```

```
Choose Login type:
0- exit
1- Admin
2- Customre
Choice: 2
Adobe Photoshop 6.0 29.99 1

Adobe Flash 3.3 9.99 1

Adobe Photoshop 6.2 39.99 1

Norton 4.5 19.99 1

Norton 5.5 29.99 1

avira 49.99 5

choose what you want to do:
0- up
1- Buy a product
choice:
```

```
choose what you want to do:
0- up
1- Buy a product
choice: 1
Enter the product name: avira
product purchesed successfully
Adobe Photoshop 6.0 29.99 1

Adobe Flash 3.3 9.99 1

Adobe Photoshop 6.2 39.99 1

Norton 4.5 19.99 1

Norton 5.5 29.99 1

avira 49.99 4
```

```
choose what you want to do:
0- up
1- add a product
2- remove a product
 3- update a product
choice: 2
Adobe Photoshop 6.0 29.99 1

Adobe Flash 3.3 9.99 1

Adobe Photoshop 6.2 39.99 1

Norton 4.5 19.99 1

Norton 5.5 29.99 1

avira 49.99 1

enter the name of the product: avira
avira
```