# GIT Department of Computer Engineering

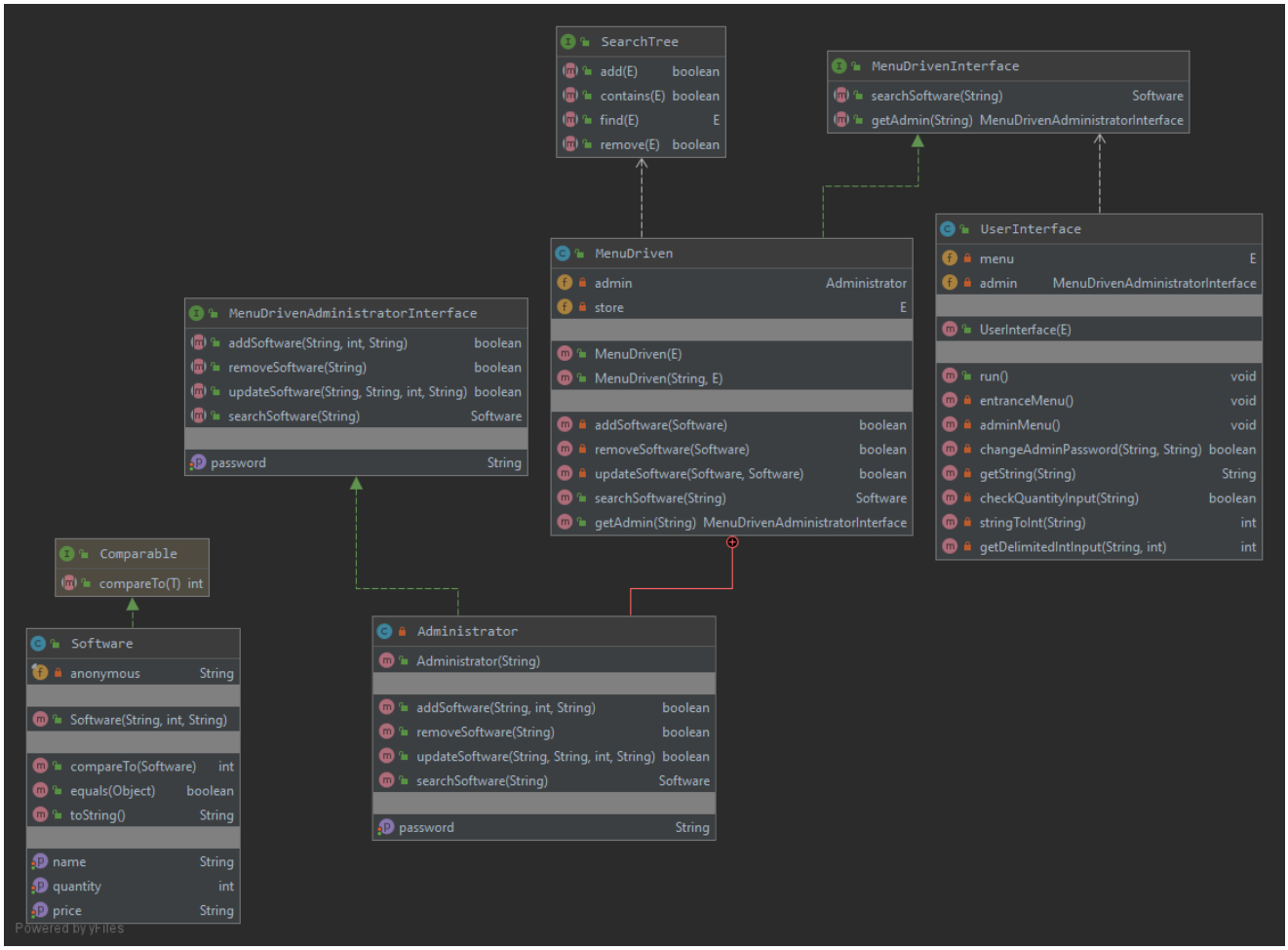
# CSE 222/505 – Spring 2020

# Homework #07 Part 4 Report

**Abdullah ÇELİK**

**171044002**

**Class Diagram**




**Problem Solution Approach**

First, some feature in the system are not public features. For this reason, I added the Administrator class internally to the Menu Driven class. The methods of the Administrator class have methods that are not available to normal users and these methods call private methods in my upper class, the Menu Driven class. In this way, I have set the features as public or private. In order for the Administrator to use these methods, the Menu Driven class must first call the getAdmin method. While doing this, if the password is not entered correctly, the administrator cannot be logged on, so that the features cannot be accessed without authority.

Secondly, the system can be used with the class that implements any SearchTree interface. For this reason, adding, deleting and updating methods take log(n) time. In the desired system, the search method has to be done depending on the name, quantity and price. It is not possible to search depending on quantity and price. Becase name is key in my system in the tree. The reason for this is that it is not possible to find because of the name as a key in the tree and therefore, the quantity and the price will be placed in the tree randomly. Since the quantity and price are not key, different named software can be added to the tree in the same amount or at the same price. There are actually ways to search quantity and price. For example, if we give each software a unique code, we will find all the unique codes when searching and see if it matches the item searched. The disadvantages of this management are the following. First, adding takes log(n) time. There is no problem here, but the find method can find the data sought to access all unique ids first. This nlog(n) takes time. Remove method should use the find method first and remove the found item from the

tree again. This means $n^2\log(n)$ time. The update method must find the same way first. This means nlog(n) time. I did not add these two methods to the system because I thought it would not make sense to use the SearchTree interface if I would use this path, and I think it would not make sense to search for software that is 15 or $ 150 for the user.

**Test Cases**

| Menu Driven will be test by using BinarySearchTree implementation(in the book) | | | | | |
|---|---|---|---|---|---|
| **Test ID** | **Scenerio** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TEST01 | One paremeter constructor called | data: bst | Successfully created menu driven system | As expected | Pass |
| TEST02 | Two parameter constructor called | password: psw<br>data: bst | Successfully created menu driven system | As expected | Pass |
| TEST03 | Testing addSoftware method when administrator password is wrong | password: wrongpsw | Login process couldn't be done | As expected | Pass |
| TEST04 | Testing addSoftware method when administrator password is correct and system doesn't contain software to add | password: psw<br>Name: Office 365<br>Quantity: 5<br>Price: 100<br>Name: Adobe Photoshop 6.0<br>Quantity: 10<br>Price: 200<br>Name: Adobe Photoshop 6.2<br>Quantity: 5<br>Price: 100<br>Name: Skype<br>Quantity: 3<br>Price: 15 | Successfully added | As expected | Pass |
| TEST05 | Testing addSoftware method when administrator password is correct and system contains software to add | password: psw<br>Name: Office 365<br>Quantity: 5<br>Price: 100 | Successfully didn't add | As expected | Pass |
| TEST06 | Testing searchSoftware method when system contains software | Name: Adobe Photoshop 6.0 | Successfully found | As expected | Pass |
| TEST07 | Testing searchSoftware method when system doesn't contain software | Name: Norton 4.5 | Successfully didn't find | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| TEST08 | Testing removeSoftware method when administrator password is wrong | password: wrongpsw | Login process couldn't be done | As expected | Pass |
| TEST09 | Testing removeSoftware method when administrator password is correct and system contains software | password: psw<br>Name: Adobe Photoshop 6.0<br>Name: Adobe Photoshop 6.2 | Successfully removed | As expected | Pass |
| TEST10 | Testing removeSoftware method when administrator password is correct and system doesn't contain software | password: psw<br>Name: Netflix | Successfully didn't remove | As expected | Pass |
| TEST11 | Testing updateSoftware method when administrator password is correct and system contains software | password: psw<br>Old software: Office 365<br><br>New software: Office 365<br>Quantity: 1<br>Price: 1 | Successfully updated | As expected | Pass |
| TEST12 | Testing updateSoftware method when administrator password is correct and system doesn't contain software | password: pst<br>Old software: Adobe Flash<br><br>New software: Adobe Flash<br>Quantity: 1<br>Price: 1 | Successfully didn't update | As expected | Pass |
| TEST13 | Testing updateSoftware method when administrator password is correct and system contains software | password: psw<br>Old software: Office 365<br><br>New software: Office 365<br>Quantity: 0<br>Price: 10 | Succesfully removed | As expected | Pass |

**Running and Results**

```
TEST01
Constructor with one parameter will be tested. Constructor will be called as
        new MenuDriven(bst)
Menu Driven was created successfully!
TEST02
Constructor with two parameter will be tested. Constructor will be called as
        new MenuDriven("psw",bst)
Menu Driven was created successfully!
```

```
Testing boolean addSoftware(String name, int quantity, String price)
This feature isn't public. So, firstly getAdmin method will be called
If password is true, adding process is start
If password is false, getAdmin method returns null

TEST03
When administrator password is wrong, method will be called as
        menu.getAdmin("worngpsw")
Login process isn't successful! Password is wrong

TEST04
When administrator password is correct and system doesn't include software to add
Firstly, getAdmin("psw") method will be called
Secondly, addSoftware method will be called respectively as
        admin.addSoftware("Office 365",5,"$100"),admin.addSoftware("Adobe Photoshop 6.0",10,"$200"),
        admin.addSoftware("Adobe Photoshop 6.2",5,"$100"),admin.addSoftware("Skype",3,"$15")
Calling administrator is successful! Password is correct

Before adding, bst:
-

Adding process for Office 365: true
Adding process for Adobe Photoshop 6.0: true
Adding process for Adobe Photoshop 6.2: true
Adding process for Skype: true

After adding, bst:
Name: Office 365, Quantity: 5, Price: 100
  Name: Adobe Photoshop 6.0, Quantity: 10, Price: $200
    -
    Name: Adobe Photoshop 6.2, Quantity: 5, Price: $100
      -
      -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

TEST05
When administrator password is correct and system includes software to add
Firstly, getAdmin("psw") method will be called
Secondly, addSoftware method will be called as admin.addSoftware("Office 365",5,"$100")
Calling administrator is successful! Password is correct

Before adding, bst:
Name: Office 365, Quantity: 5, Price: 100
  Name: Adobe Photoshop 6.0, Quantity: 10, Price: $200
    -
    Name: Adobe Photoshop 6.2, Quantity: 5, Price: $100
      -
      -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Adding process: false

After adding, bst:
Name: Office 365, Quantity: 5, Price: 100
  Name: Adobe Photoshop 6.0, Quantity: 10, Price: $200
    -
    Name: Adobe Photoshop 6.2, Quantity: 5, Price: $100
      -
      -
  Name: Skype, Quantity: 3, Price: $15
    -
    -
```

```
TEST06
When system contains software, method will be called as
        menu.searchSoftware("Adobe Photoshop 6.0")
Finding: Name: Adobe Photoshop 6.0, Quantity: 10, Price: $200

TEST07
When system doesn't contain software, method will be called as
        menu.searchSoftware("Norton 4.5")
Finding: null

TEST08
When administrator password is wrong, method will be called as
        menu.getAdmin("worngpsw")
Login process isn't successful! Password is wrong

TEST09
When administrator password is correct and system contains software,
method will be called respectively as
        admin.removeSoftware("Adobe Photoshop 6.0"), admin.removeSoftware("Adobe Photoshop 6.2")
Calling administrator is successful! Password is correct

Before removing, bst:
Name: Office 365, Quantity: 5, Price: 100
  Name: Adobe Photoshop 6.0, Quantity: 10, Price: $200
    -
    Name: Adobe Photoshop 6.2, Quantity: 5, Price: $100
      -
      -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Removing Adobe Photoshop 6.0: true

After removing, bst:
Name: Office 365, Quantity: 5, Price: 100
  Name: Adobe Photoshop 6.2, Quantity: 5, Price: $100
    -
    -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Removing Adobe Photoshop 6.0: true

After removing, bst:
Name: Office 365, Quantity: 5, Price: 100
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -
```

```
TEST10
When administrator password is correct and system doesn't contain software,
method will be called respectively as
        admin.removeSoftware("Netflix")

Before removing, bst:
Name: Office 365, Quantity: 5, Price: 100
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Removing Adobe Photoshop 6.0: false

After removing, bst:
Name: Office 365, Quantity: 5, Price: 100
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

TEST11
When administrator password is true and system contains software, method will be called as
        admin.updateSoftware("Office 365","Office 365",1,"$1")
Calling administrator is successful! Password is correct

Before updating, bst:
Name: Office 365, Quantity: 5, Price: 100
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Updating process: false

After updating, bst:
Name: Office 365, Quantity: 1, Price: $1
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -
```

```
TEST12
When administrator password is true and system doesn't contain software, method will be called as
        admin.updateSoftware("Adobe Flash","Adobe Flash",1,"$1")

Before updating, bst:
Name: Office 365, Quantity: 1, Price: $1
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Updating process: false

After updating, bst:
Name: Office 365, Quantity: 1, Price: $1
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

TEST13
When administrator password is true and system contains software, method will be called as
        admin.updateSoftware("Office 365","Office 365",0,$10)

Before updating, bst:
Name: Office 365, Quantity: 1, Price: $1
  -
  Name: Skype, Quantity: 3, Price: $15
    -
    -

Updating process: false

After updating, bst:
Name: Skype, Quantity: 3, Price: $15
  -
  -
```