

## - Admin

```
public boolean addBranch(int branchCode){
    Branch br = new Branch(getCompany(), branchCode);  $\theta(1)$ 
    return getCompany().getBranches().add(br);  $Tb = O(1), Tw = O(n)$  }  $O(n)$ 
}
```

```
public boolean removeBranch(int branchCode){
    Branch br = new Branch(getCompany(), branchCode);  $\theta(1)$ 
    return getCompany().getBranches().remove(br);  $Tb = O(1), Tw = O(n)$  }  $O(n)$ 
}
```

```
public boolean addBranchEmployee(int branchCode, String name, String surname, String email, String password){
    Branch br = new Branch(getCompany(), branchCode);  $\theta(1)$ 
    int index = getCompany().getBranches().indexOf(br);  $Tb = O(1), Tw = O(n)$ 
    if (index == -1)  $\theta(1)$ 
        return false;
    BranchEmployee be = new BranchEmployee(name, surname, email, password, br);  $\theta(1)$ 
    return getCompany().getEmployees().add(be);  $Ta = \theta(1)$  }  $= O(n)$ 
}
```

```
public boolean removeBranchEmployee(int branchCode, int userID){
    Branch br = new Branch(getCompany(), branchCode);  $\theta(1)$ 
    int index = getCompany().getBranches().indexOf(br);  $Tb = O(1), Tw = O(n)$ 
    if (index == -1)  $\theta(1)$ 
        return false;
    BranchEmployee be = new BranchEmployee(br, userID);  $\theta(1)$ 
    return getCompany().getEmployees().remove(be);  $\theta(n)$  }  $= O(n)$ 
}
```

```
public void printBranches(){
    MSLinkedList<Branch> bList = company.getBranches();  $\theta(1)$ 
    Iterator<Branch> iter = bList.iterator();  $\theta(1)$ 
    while (iter.hasNext()){
        Branch branch = iter.next();  $\theta(1)$ 
        System.out.println(branch.toString());  $\theta(1)$  }  $\theta(n)$  }  $= \theta(n)$ 
}
```

```
public void printEmployees(){
    int len = getCompany().getEmployees().size();  $\theta(1)$ 
    for (int i=0; i<len; i++){
        System.out.println(getCompany().getEmployees().get(i).toString());  $\theta(1)$  }  $\theta(n)$  }  $= \theta(n)$ 
}
```

## - Branch Employee

```
public boolean addUser(String name, String surname, String email, String password){
    Customer cs = new Customer(name, surname, email, password);  $\theta(1)$ 
    return getBranch().getCompany().getCustomers().add(cs);  $Ta = \theta(1)$   $= \theta(1)$ 
}
```

```
public boolean removeUser(int userID){
    Customer cs = new Customer(userID);  $\theta(1)$ 
    return getBranch().getCompany().getCustomers().remove(cs);  $\theta(n)$   $= \theta(n)$ 
}
```

```
public void printPreviousOrders(int userID){
    Customer cs = new Customer(userID);  $\theta(1)$ 
    int index = getBranch().getCompany().getCustomers().indexOf(cs);  $O(n)$ 
    if (index == -1)
    {
        System.out.println("Invalid ID");  $\theta(1)$ 
        return;
    }

    getBranch().getCompany().getCustomers().get(index).printPreviousOrders();  $\theta(n)$ 
}  $= O(n)$ 
```

```
public void addProduct(Furniture furniture){
    if (furniture instanceof OfficeDesk){
        int index = getBranch().getCompany().getFurniture().get(0).indexOf(furniture);  $O(n)$ 
        if (index != -1){
            Furniture fr = getBranch().getCompany().getFurniture().get(0).get(index);  $\theta(1)$ 
            if (fr.getBranchCode() == furniture.getBranchCode()){
                System.out.println("The product already exists!");  $\theta(1)$ 
                return;
            }
        }
        getBranch().getCompany().getFurniture().get(0).add(furniture);  $Ta = \theta(1)$ 
    }

    else if(furniture instanceof OfficeChair){
        int index = getBranch().getCompany().getFurniture().get(1).indexOf(furniture);
        if (index != -1){
            Furniture fr = getBranch().getCompany().getFurniture().get(1).get(index);
            if (fr.getBranchCode() == furniture.getBranchCode()){
                System.out.println("The product already exists!");
                return;
            }
        }
        getBranch().getCompany().getFurniture().get(1).add(furniture);
    }

    else if(furniture instanceof MeetingTable){
        int index = getBranch().getCompany().getFurniture().get(2).indexOf(furniture);
        if (index != -1){
            Furniture fr = getBranch().getCompany().getFurniture().get(2).get(index);
            if (fr.getBranchCode() == furniture.getBranchCode()){
                System.out.println("The product already exists!");
                return;
            }
        }
    }
}
```

```
public boolean removeProduct(Furniture furniture){

    if (furniture instanceof OfficeDesk)
        return getBranch().getCompany().getFurniture().get(0).remove(furniture);

    else if(furniture instanceof OfficeChair)
        return getBranch().getCompany().getFurniture().get(1).remove(furniture);  $= O(n)$ 

    else if(furniture instanceof MeetingTable)
        return getBranch().getCompany().getFurniture().get(2).remove(furniture);

    return false;
}
```

```

public int getStock(Furniture furniture){
    if (furniture instanceof OfficeDesk){
        int index = getBranch().getCompany().getFurniture().get(0).indexOf(furniture); O(n)
        if (index == -1){
            return 0;
        }
        return getBranch().getCompany().getFurniture().get(0).get(index).getStock(); O(1)
    }
}

```

**= O(n)**

## - Customer

```

public void listProducts(Company company, int branchCode){
    HybridList<Furniture> hL = company.getFurniture(); θ(1)
    Iterator<MSArrayList<Furniture>> iter = hL.iterator(); θ(1)
    while (iter.hasNext()){
        MSArrayList<Furniture> fur = iter.next(); θ(1)
        for (int i=0; i<fur.size(); i++){
            if (fur.get(i).getBranchCode() == branchCode){ θ(1)
                fur.get(i).print(); θ(1)
                System.out.println(" Branch: " + fur.get(i).getBranchCode()); θ(1)
            }
        }
    }
}

```

**} θ(n) } θ(n<sup>2</sup>)**

```

public void printPreviousOrders(){
    if (oldOrder.isEmpty())
        System.out.println("Cart is empty!"); θ(1)
    else {
        int len = oldOrder.size();
        System.out.println("Order History: "); θ(1)
        for (int i=0; i<len; i++)
            oldOrder.get(i).print(); θ(1)
    }
}

```

**= θ(n)**