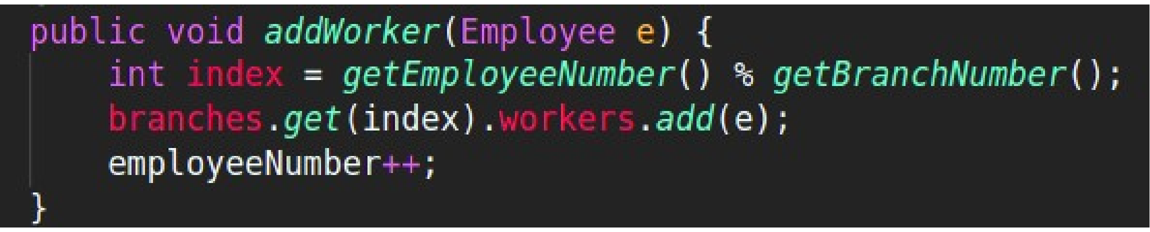
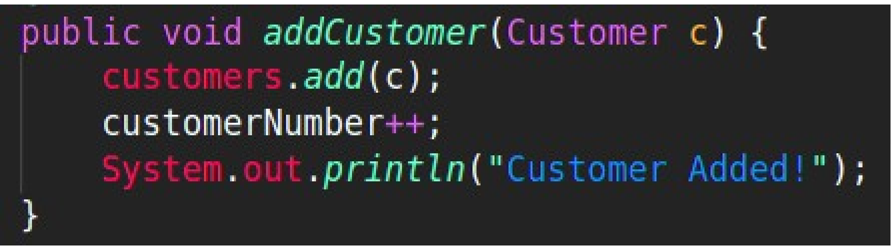


```java
public void addWorker(Employee e) {
    int index = getEmployeeNumber() % getBranchNumber();
    branches.get(index).workers.add(e);
    employeeNumber++;
}
```

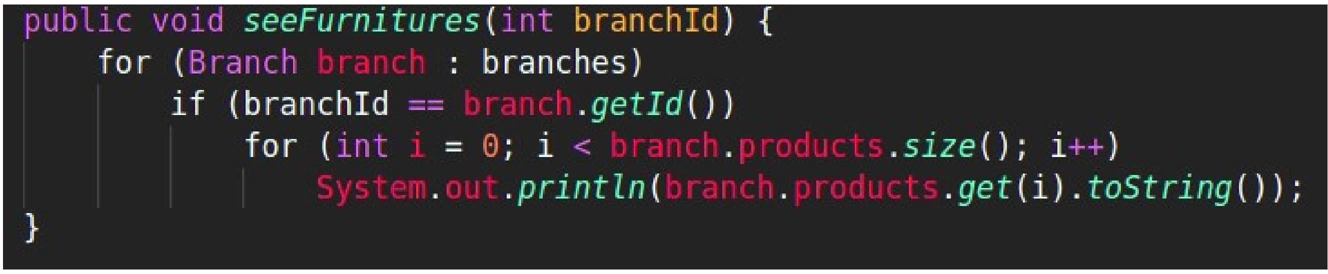
Θ(1)
Θ(n)->get
Total: Θ(n)



```java
public void addCustomer(Customer c) {
    customers.add(c);
    customerNumber++;
    System.out.println("Customer Added!");
}
```

Θ(1)

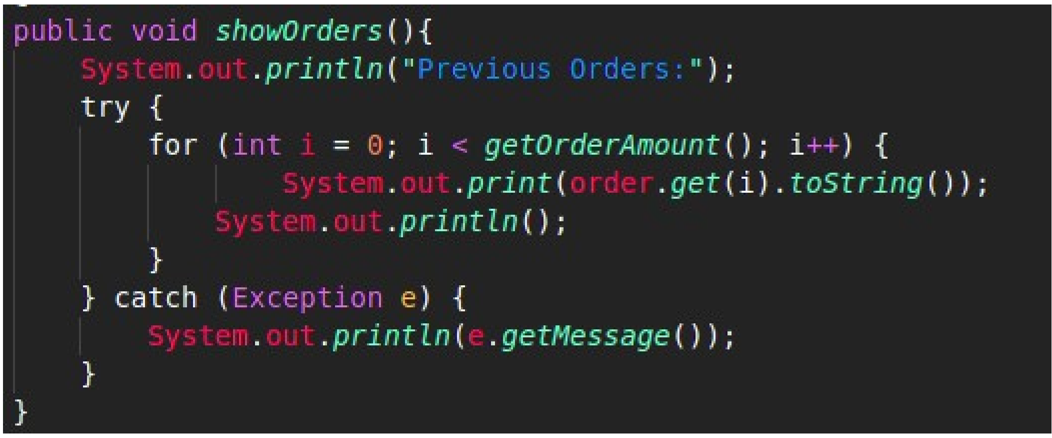
Total:Θ(1)



```java
public void seeFurnitures(int branchId) {
    for (Branch branch : branches)
        if (branchId == branch.getId())
            for (int i = 0; i < branch.products.size(); i++)
                System.out.println(branch.products.get(i).toString());
}
```

Θ(n)

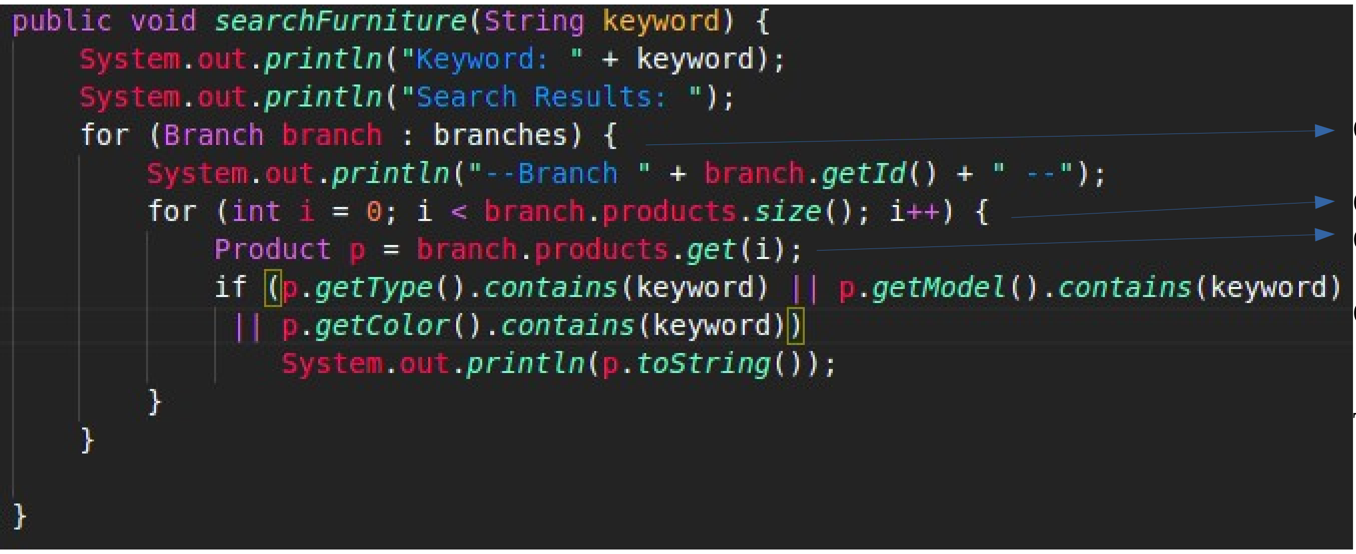
Θ(k)
Θ(x)
Total:Θ(nkx)



```java
public void showOrders(){
    System.out.println("Previous Orders:");
    try {
        for (int i = 0; i < getOrderAmount(); i++) {
            System.out.print(order.get(i).toString());
            System.out.println();
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

Θ(n)
Θ(k)->get, Θ(1)->tostring

Total:Θ(n+k)



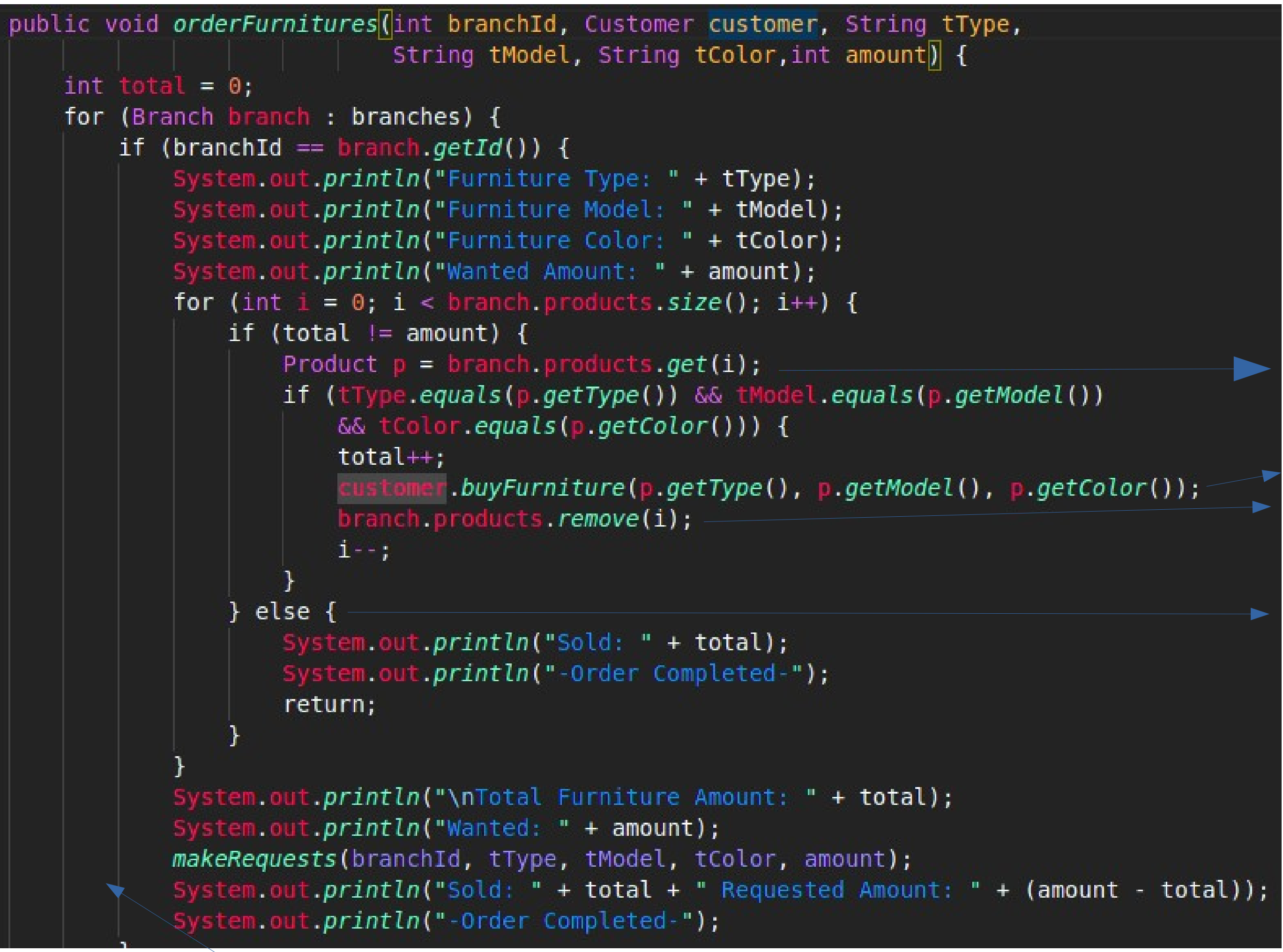
```java
public void searchFurniture(String keyword) {
    System.out.println("Keyword: " + keyword);
    System.out.println("Search Results: ");
    for (Branch branch : branches) {
        System.out.println("--Branch " + branch.getId() + " --");
        for (int i = 0; i < branch.products.size(); i++) {
            Product p = branch.products.get(i);
            if (p.getType().contains(keyword) || p.getModel().contains(keyword)
              || p.getColor().contains(keyword))
                System.out.println(p.toString());
        }
    }

}
```

Θ(n)

Θ(k)
Θ(m)

Θ(1)

Total:Θ(nkm)



```java
public void orderFurnitures(int branchId, Customer customer, String tType,
                            String tModel, String tColor,int amount) {
    int total = 0;
    for (Branch branch : branches) {
        if (branchId == branch.getId()) {
            System.out.println("Furniture Type: " + tType);
            System.out.println("Furniture Model: " + tModel);
            System.out.println("Furniture Color: " + tColor);
            System.out.println("Wanted Amount: " + amount);
            for (int i = 0; i < branch.products.size(); i++) {
                if (total != amount) {
                    Product p = branch.products.get(i);
                    if (tType.equals(p.getType()) && tModel.equals(p.getModel())
                        && tColor.equals(p.getColor())) {
                        total++;
                        customer.buyFurniture(p.getType(), p.getModel(), p.getColor());
                        branch.products.remove(i);
                        i--;
                    }
                } else {
                    System.out.println("Sold: " + total);
                    System.out.println("-Order Completed-");
                    return;
                }
            }
            System.out.println("\nTotal Furniture Amount: " + total);
            System.out.println("Wanted: " + amount);
            makeRequests(branchId, tType, tModel, tColor, amount);
            System.out.println("Sold: " + total + " Requested Amount: " + (amount - total));
            System.out.println("-Order Completed-");
```

$\Theta(n)$

$\Theta(k)$

$\Theta(m)$

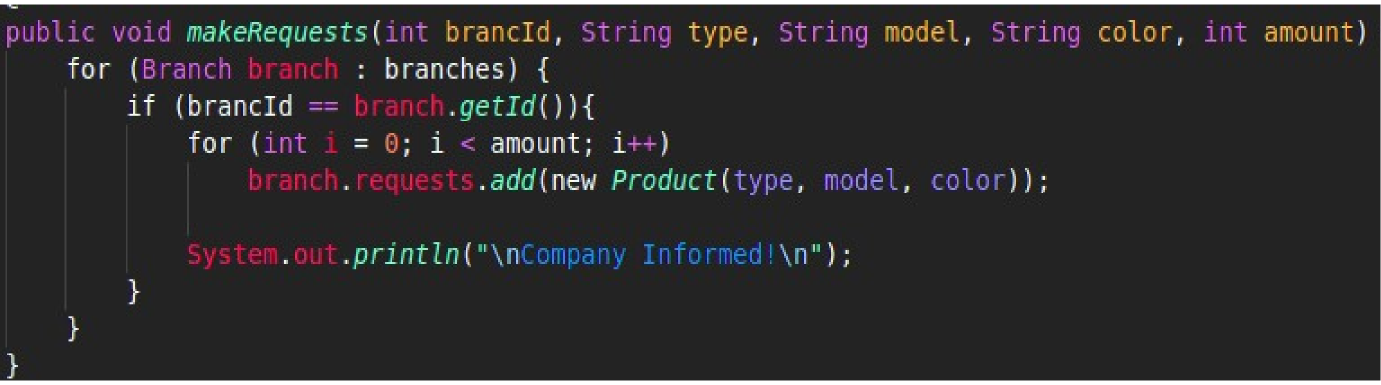
$\Theta(p)$

$\Theta(x)$

$\Theta(1)$

$\Theta(abc)$

$T_{average}$:$\Theta((k(m+p+x))+abc)$



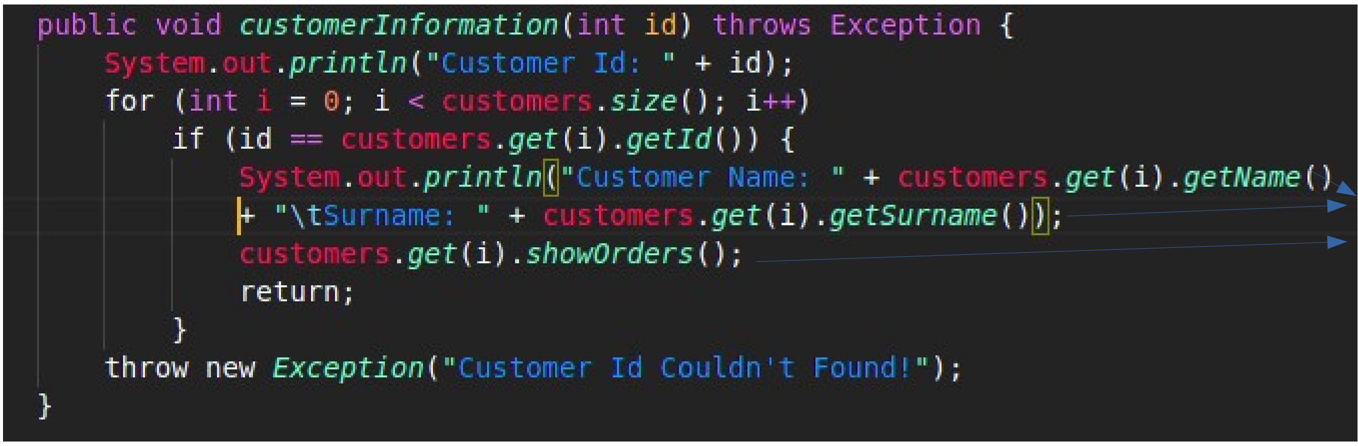
```java
public void makeRequests(int brancId, String type, String model, String color, int amount)
    for (Branch branch : branches) {
        if (brancId == branch.getId()){
            for (int i = 0; i < amount; i++)
                branch.requests.add(new Product(type, model, color));

            System.out.println("\nCompany Informed!\n");
        }
    }
}
```

$\Theta(a)$

$\Theta(b)$

$\Theta(c)$

Total:$\Theta(abc)$



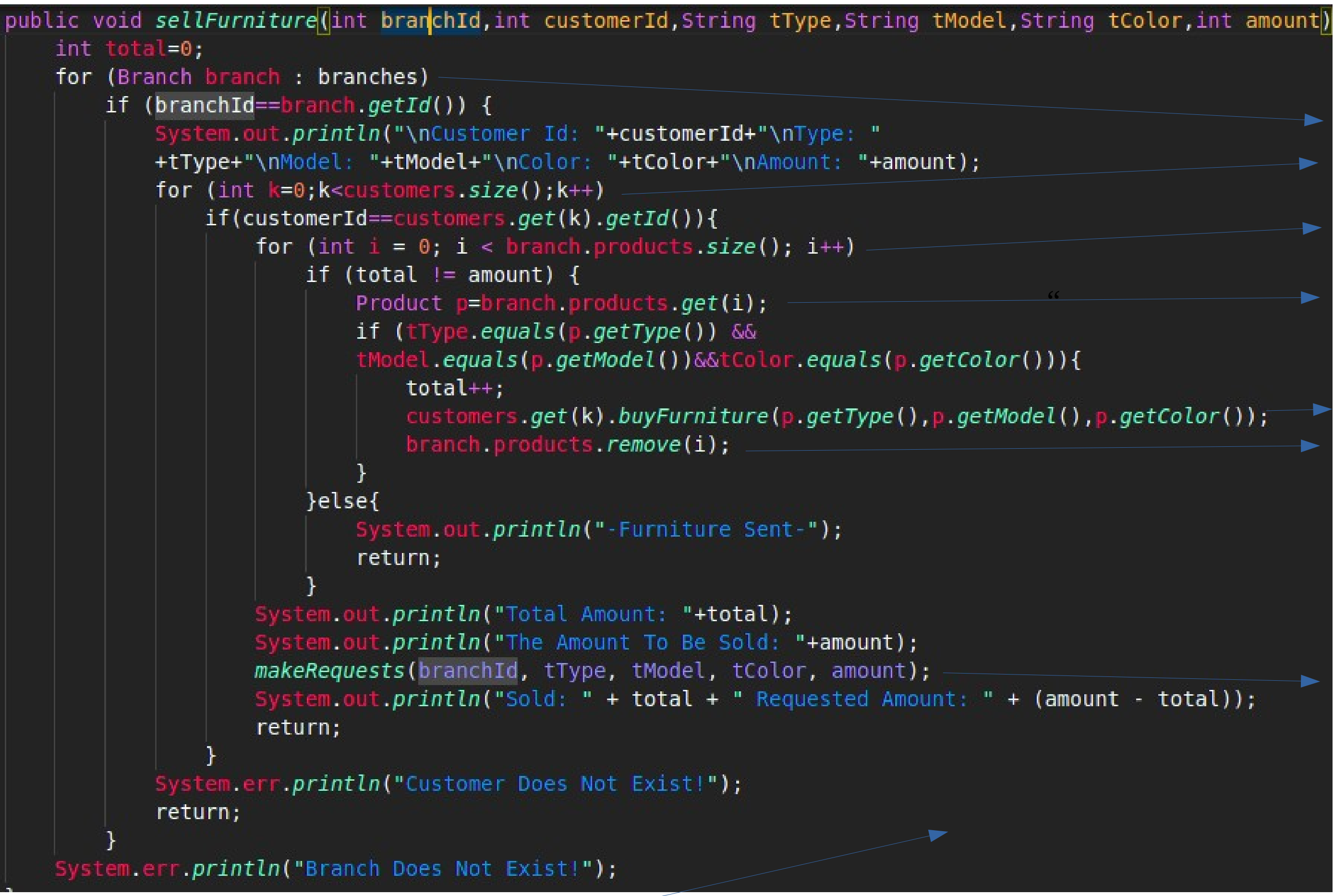
```java
public void customerInformation(int id) throws Exception {
    System.out.println("Customer Id: " + id);
    for (int i = 0; i < customers.size(); i++)
        if (id == customers.get(i).getId()) {
            System.out.println("Customer Name: " + customers.get(i).getName()
            + "\tSurname: " + customers.get(i).getSurname());
            customers.get(i).showOrders();
            return;
        }
    throw new Exception("Customer Id Couldn't Found!");
}
```

$\Theta(n)$

$\Theta(1)$

$\Theta(k)$

Total:$\Theta(nk)$



```
public void sellFurniture(int branchId,int customerId,String tType,String tModel,String tColor,int amount)
    int total=0;
    for (Branch branch : branches)
        if (branchId==branch.getId()) {
            System.out.println("\nCustomer Id: "+customerId+"\nType: "
            +tType+"\nModel: "+tModel+"\nColor: "+tColor+"\nAmount: "+amount);
            for (int k=0;k<customers.size();k++)
                if(customerId==customers.get(k).getId()){
                    for (int i = 0; i < branch.products.size(); i++)
                        if (total != amount) {
                            Product p=branch.products.get(i);
                            if (tType.equals(p.getType()) &&
                            tModel.equals(p.getModel())&&tColor.equals(p.getColor())){
                                total++;
                                customers.get(k).buyFurniture(p.getType(),p.getModel(),p.getColor());
                                branch.products.remove(i);
                            }
                        }else{
                            System.out.println("-Furniture Sent-");
                            return;
                        }
                    System.out.println("Total Amount: "+total);
                    System.out.println("The Amount To Be Sold: "+amount);
                    makeRequests(branchId, tType, tModel, tColor, amount);
                    System.out.println("Sold: " + total + " Requested Amount: " + (amount - total));
                    return;
                }
            System.err.println("Customer Does Not Exist!");
            return;
        }
    System.err.println("Branch Does Not Exist!");
}
```

$\Theta(n)$

$\Theta(k)$

$\Theta(j)$

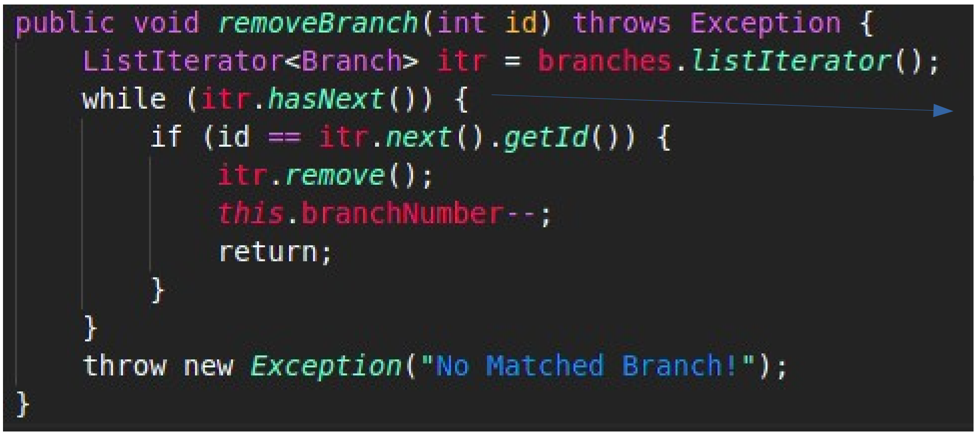
$\Theta(m)$

$\Theta(p)$

$\Theta(x)$

$\Theta(abc)$

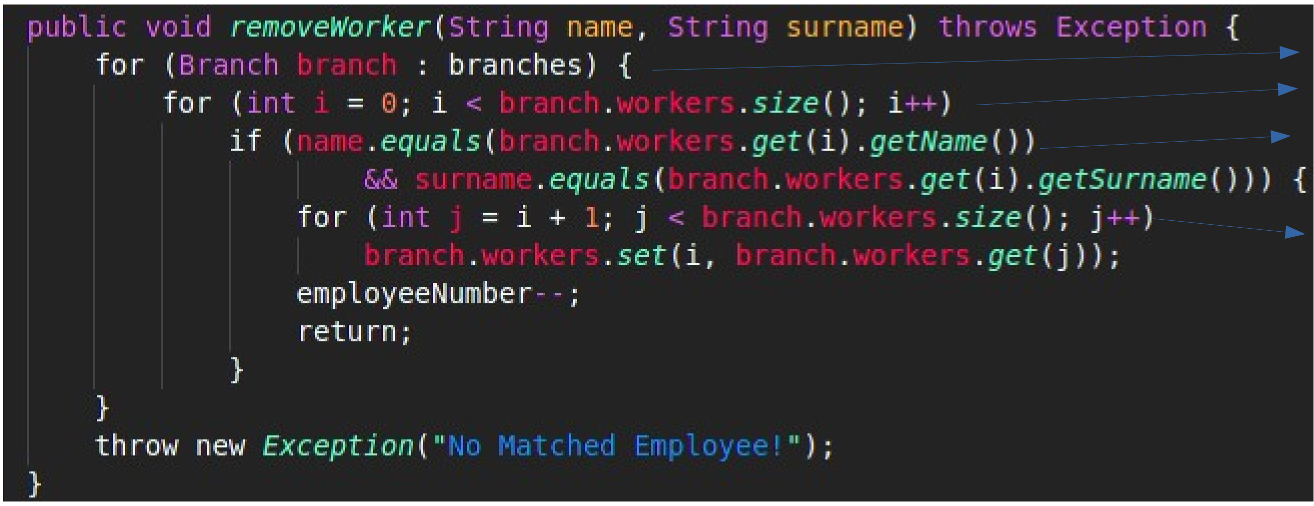
$T_{average}$:$\Theta(n(k(j(m+p+x)+abc)))$



```
public void removeBranch(int id) throws Exception {
    ListIterator<Branch> itr = branches.listIterator();
    while (itr.hasNext()) {
        if (id == itr.next().getId()) {
            itr.remove();
            this.branchNumber--;
            return;
        }
    }
    throw new Exception("No Matched Branch!");
}
```

$\Theta(n)$

Total:$\Theta(n)$



```
public void removeWorker(String name, String surname) throws Exception {
    for (Branch branch : branches) {
        for (int i = 0; i < branch.workers.size(); i++)
            if (name.equals(branch.workers.get(i).getName())
                    && surname.equals(branch.workers.get(i).getSurname())) {
                for (int j = i + 1; j < branch.workers.size(); j++)
                    branch.workers.set(i, branch.workers.get(j));
                employeeNumber--;
                return;
            }
    }
    throw new Exception("No Matched Employee!");
}
```

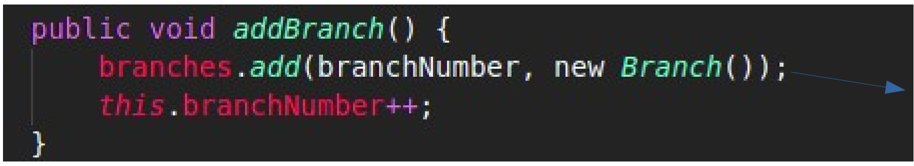
$\Theta(n)$

$\Theta(k)$

$\Theta(1)$

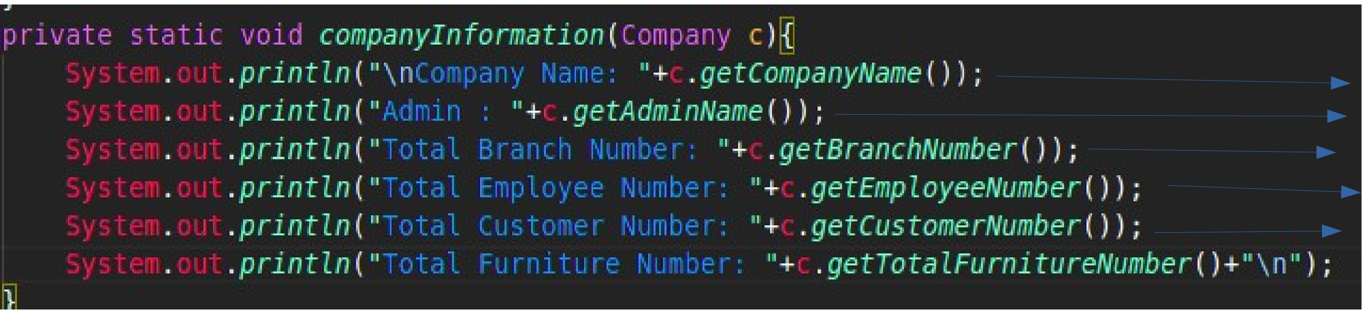
$\Theta(k-i)$

Total: $\Theta(nk)$



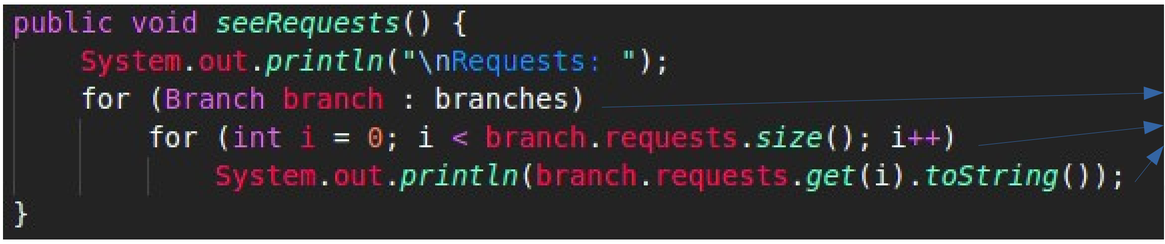
```
public void addBranch() {
    branches.add(branchNumber, new Branch());
    this.branchNumber++;
}
```

Θ(n)
Total: Θ(n)



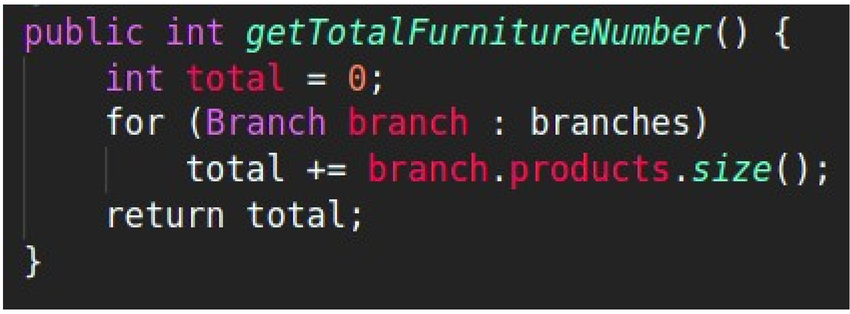
```
private static void companyInformation(Company c){
    System.out.println("\nCompany Name: "+c.getCompanyName());
    System.out.println("Admin : "+c.getAdminName());
    System.out.println("Total Branch Number: "+c.getBranchNumber());
    System.out.println("Total Employee Number: "+c.getEmployeeNumber());
    System.out.println("Total Customer Number: "+c.getCustomerNumber());
    System.out.println("Total Furniture Number: "+c.getTotalFurnitureNumber()+"\n");
}
```

Θ(1)
Θ(1)
Θ(1)
Θ(1)
Θ(1)
Θ(n)
Total:Θ(n)



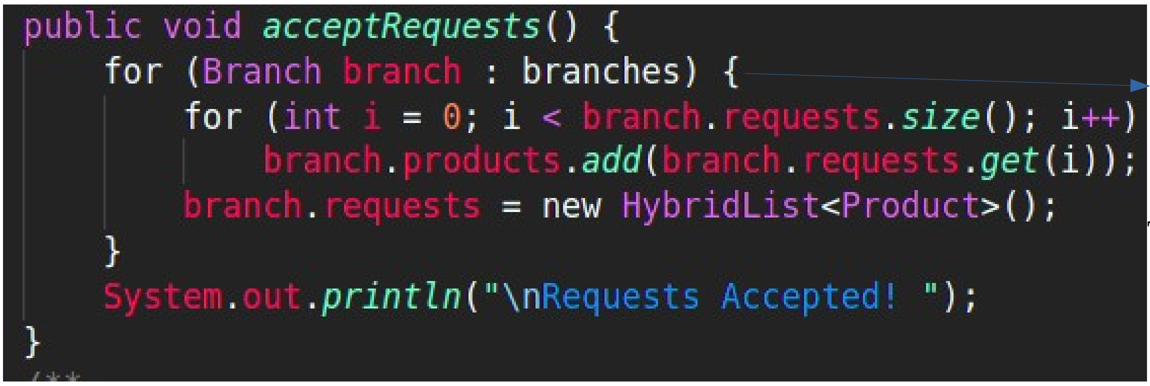
```
public void seeRequests() {
    System.out.println("\nRequests: ");
    for (Branch branch : branches)
        for (int i = 0; i < branch.requests.size(); i++)
            System.out.println(branch.requests.get(i).toString());
}
```

Θ(n)
Θ(k)
Θ(m)
Total: Θ(nkm)



```
public int getTotalFurnitureNumber() {
    int total = 0;
    for (Branch branch : branches)
        total += branch.products.size();
    return total;
}
```

Θ(m)
Θ(1)
Total: Θ(m)



```
public void acceptRequests() {
    for (Branch branch : branches) {
        for (int i = 0; i < branch.requests.size(); i++)
            branch.products.add(branch.requests.get(i));
        branch.requests = new HybridList<Product>();
    }
    System.out.println("\nRequests Accepted! ");
}
```

Θ(n)
Θ(k)
Θ(m)-> add ,Θ(p)->get

Total: Θ(nk(m+p))

*All the methods of admin class ,calls company class' methods.
**The time complexities of all the methods that employee class has, are Θ(1).
***The methods of customer class that is not here, have Θ(1) time complexity.

* The time complexities of all the methods that branch class has, are Θ(1).
**The methods of company class that is not here, have Θ(1) time complexity.

*The time complexities of all the methods that product class has, are Θ(1).
**The time complexities of all the methods that classes which are instance of furniture(Meeting Table,Office Chair vs.) , are Θ(1).