"Experiment 2.2"

Student Name: **SUMIT KUMAR** UID: **20BCS8226**
Branch: **CSE** Section/Group: **808-A**
Semester: **5** Date of Submission: **28-10-22**
Subject Name: **PBLJ Lab** Subject Code: **20CSP-321**

Aim:

Write a Program to collect and groups the card.

Minimum Hardware Requirements:

- 2 GHz CPU or 1 virtual CPU in virtualized environments.
- 1 GB of RAM.
- 4 GB of storage.

Minimum Software Requirements:

Software	Version
• OS	• Mac OS 10.15, HP-UX 11i V3, AIX 7.2, Windows Server 2019, Windows 10, Solaris 11.3, Red Hat Enterprise Linux 8.1, Ubuntu Server 20.04
• JDK	• JDK 1.8.0, JDK 11, Ellipse IDE, Net, NetBeans 8.2

Source Code:

// Save: GroupCards.java

```
// SUMIT KUMAR
// 20BCS8226

package practice2;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeMap;

public class GroupCards {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		Map<Character, ArrayList<Card>> map = new TreeMap<>();
		System.out.println("Enter Number of Cards :");
		int n = sc.nextInt();
		sc.nextLine();
		for (int i = 1; i <= n; i++) {
			System.out.println("Enter card " + n);
			char symbol = sc.nextLine().charAt(0);
			int number = sc.nextInt();
			Card card = new Card();
			card.setSymbol(symbol);
			card.setNumber(number);
			sc.nextLine();
			if (!map.containsKey(symbol)) {
				ArrayList<Card> list = new ArrayList<>();
				list.add(card);
				map.put(symbol, list);
			} else {
				ArrayList<Card> list = map.get(symbol);
				list.add(card);
			}
		}
		System.out.println("Distinct Symbols are :");
		Set<Entry<Character, ArrayList<Card>>> set = map.entrySet();
		Iterator<Entry<Character, ArrayList<Card>>> it = set.iterator();
		while (it.hasNext()) {
			System.out.print(it.next().getKey() + " ");
		}
		System.out.println();
		set = map.entrySet();
		it = set.iterator();

		while (it.hasNext()) {
			int sum = 0;
			Map.Entry<Character, ArrayList<Card>> me = it.next();
			ArrayList<Card> list = me.getValue();
			System.out.println("Cards in " + me.getKey() + " Symbol");
			for (Card card : list) {
		System.out.println(card.getSymbol() + " " + card.getNumber());
```

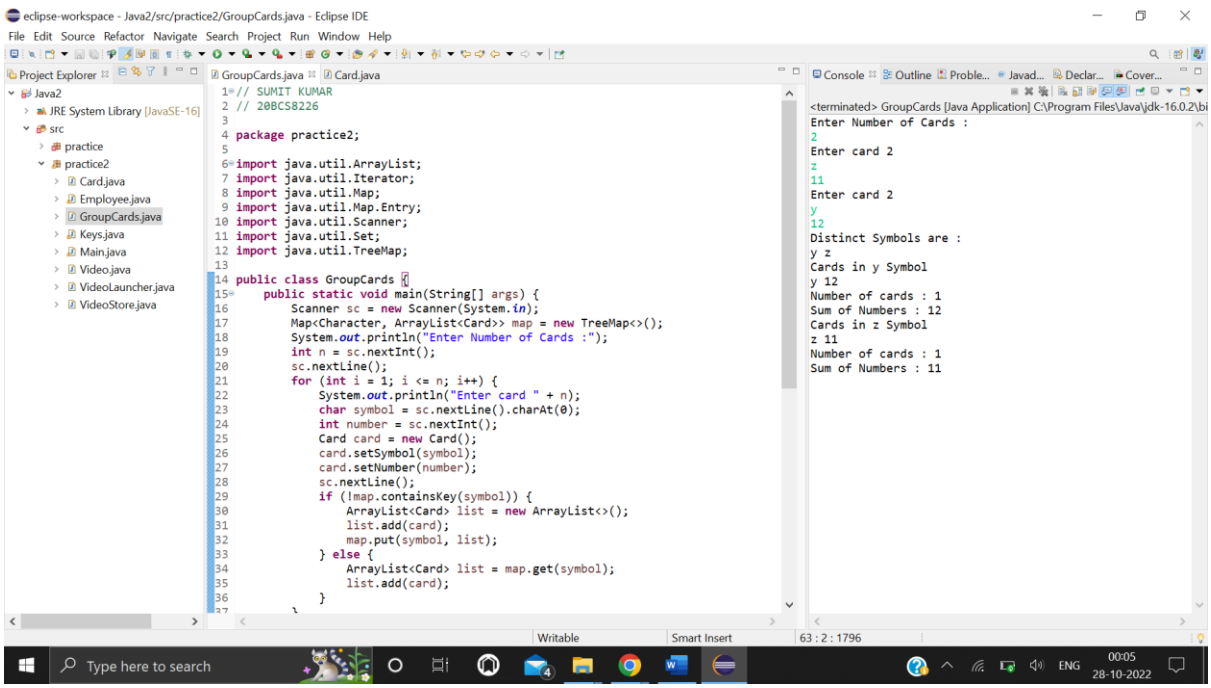
```
                sum += card.getNumber();
                }
                System.out.println("Number of cards : " + list.size());
                System.out.println("Sum of Numbers : " + sum);
            }
            sc.close();
        }

}
```

// Save: Card.java

```
package practice2;

public class Card implements Comparable<Card>
{
        private char symbol;
        private int number;
        public Card() {}
        public Card(char symbol, int number) {
                super();
                this.symbol = symbol;
                this.number = number;
        }
        public char getSymbol() {
                return symbol;
        }
        public void setSymbol(char symbol) {
                this.symbol = symbol;
        }
        public int getNumber() {
                return number;
        }
        public void setNumber(int number) {
                this.number = number;
        }



@Override
        public String toString() {
                return "Card [symbol=" + symbol + ", number=" + number + "]";
        }
        @Override
        public int compareTo(Card o) {
                if (this.symbol < o.symbol) return -1;
                else if (this.symbol > o.symbol) return 1;
                else return 1;
        }
//      @Override
//      public int hashCode() {
//          return String.valueOf(symbol).hashCode();
//      }
//      @Override
//     public boolean equals(Object obj){
//          if (obj instanceof Card) {
```

```
//            Card card = (Card) obj;
//             return (card.symbol == this.symbol);
//         } else {
//             return false;
//         }
//     }
}
```

Output:



Learning outcomes:

1. We have implemented the hash map using the java collection framework.

2. I've performed the insertion of data into hashmap using the scanner class of java.

3. I have gone through the uses of for loop for taking the input from user in iterative way.

4. Used two data types which String as key and Integer as value pair. Hashmap data structure is mainly used for insert the value in the key and value pair.