Experiment 5

Name: Mayank Sharma **UID: 22BCS16886**
Branch: BE-CSE **Section: 22BCS_IOT_EPAM_801-B**
Semester:6th **Date of Performance: 24 Feb 2025**
Subject Name: Project Based Learning in Java with Lab **Subject Code: 22CSH-359**

1. **Aim:** Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed.

2. **Algorithm:**

Step 1: Input Handling

1. Start the program and prompt the user to enter the number of cards (`N`).
2. Initialize a **Map (`LinkedHashMap`)** to store cards with symbols as keys and a list of `Card` objects as values.
3. Iterate `N` times to take inputs:
   - Read **symbol** as a `String`.
   - Read **number** as an `int`.
   - If the number is **non-positive**, prompt for re-entry.
   - Store the card in the map under its corresponding symbol.

Step 2: Process and Display Distinct Symbols

4. Extract all **unique symbols** from the map.
5. Display symbols **sorted in alphabetical order**.

Step 3: Display Card Details

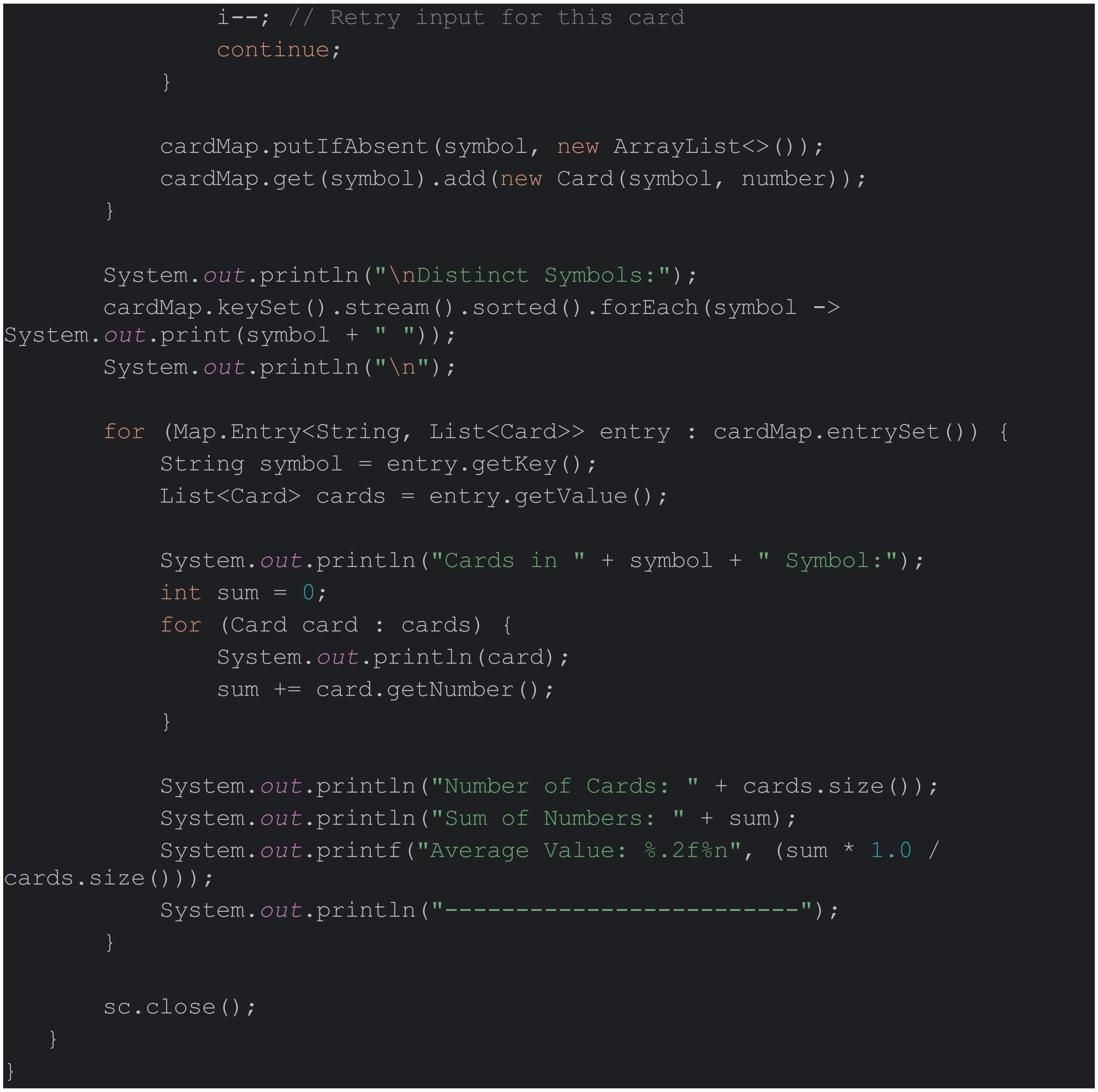
6. For each symbol:
   - Retrieve the list of cards.
   - Print each card's **symbol and number**.
   - Compute the **sum** of all numbers for the symbol.
   - Compute the **count** of cards for the symbol.
   - Compute and display the **average value** of the numbers.

Step 4: End Program

3. Close the scanner and terminate the program.

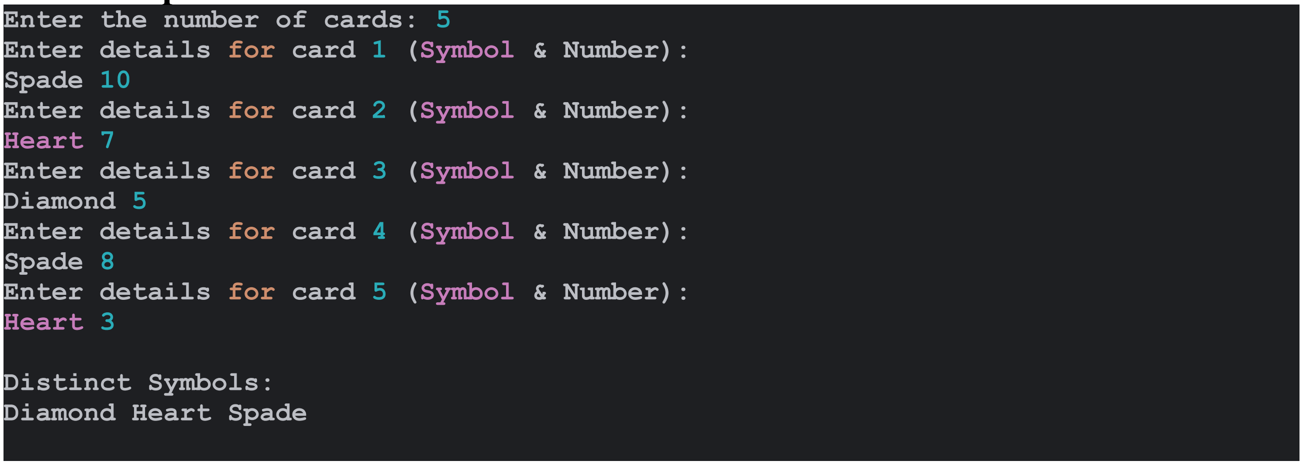
4. Implementation/Code:

```
import java.util.*;

class Card {
   private String symbol;
   private int number;

   public Card(String symbol, int number) {
       this.symbol = symbol;
       this.number = number;
   }

   public String getSymbol() {
       return symbol;
   }

   public int getNumber() {
       return number;
   }

   @Override
   public String toString() {
       return symbol + " " + number;
   }
}

public class CardCollection {
   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       Map<String, List<Card>> cardMap = new LinkedHashMap<>();

       System.out.print("Enter the number of cards: ");
       int n = sc.nextInt();
       sc.nextLine(); // Consume newline

       for (int i = 1; i <= n; i++) {
           System.out.println("Enter details for card " + i + " (Symbol &
Number): ");
           String symbol = sc.nextLine().trim();
           int number = sc.nextInt();
           sc.nextLine(); // Consume newline

           if (number <= 0) {
               System.out.println("Invalid number! Please enter a positive
value.");
```

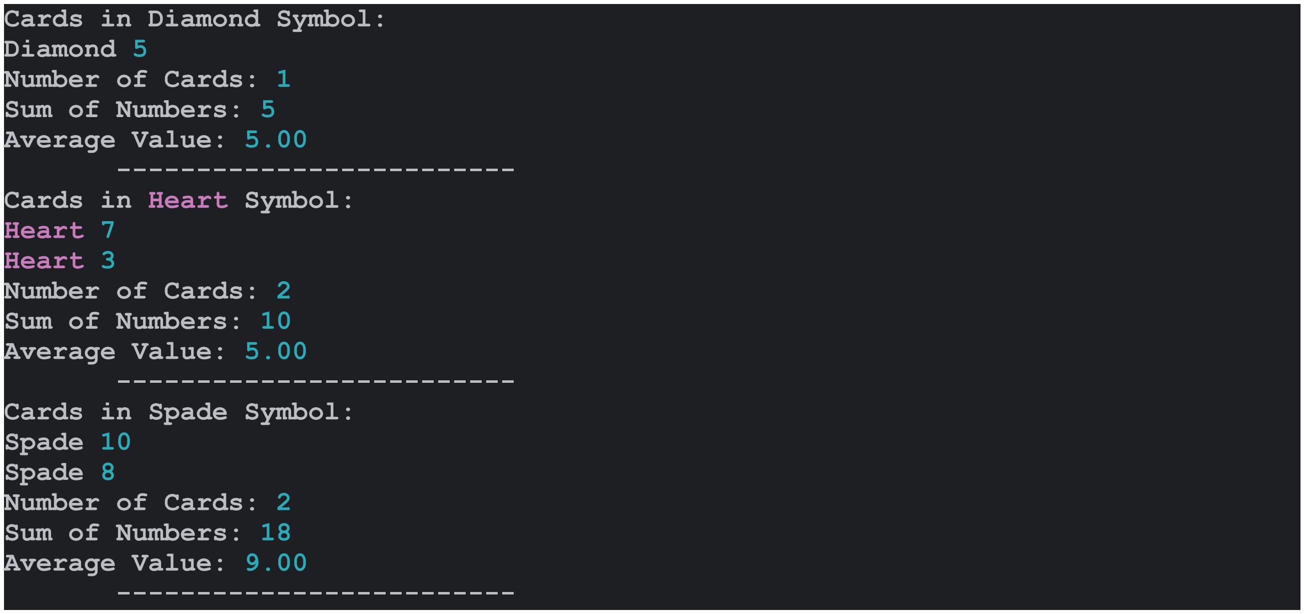


```
                i--; // Retry input for this card
                continue;
            }

            cardMap.putIfAbsent(symbol, new ArrayList<>());
            cardMap.get(symbol).add(new Card(symbol, number));
        }

        System.out.println("\nDistinct Symbols:");
        cardMap.keySet().stream().sorted().forEach(symbol ->
System.out.print(symbol + " "));
        System.out.println("\n");

        for (Map.Entry<String, List<Card>> entry : cardMap.entrySet()) {
            String symbol = entry.getKey();
            List<Card> cards = entry.getValue();

            System.out.println("Cards in " + symbol + " Symbol:");
            int sum = 0;
            for (Card card : cards) {
                System.out.println(card);
                sum += card.getNumber();
            }

            System.out.println("Number of Cards: " + cards.size());
            System.out.println("Sum of Numbers: " + sum);
            System.out.printf("Average Value: %.2f%n", (sum * 1.0 /
cards.size()));
            System.out.println("-------------------------");
        }

        sc.close();
    }
}
```

5. Output:



```
Enter the number of cards: 5
Enter details for card 1 (Symbol & Number):
Spade 10
Enter details for card 2 (Symbol & Number):
Heart 7
Enter details for card 3 (Symbol & Number):
Diamond 5
Enter details for card 4 (Symbol & Number):
Spade 8
Enter details for card 5 (Symbol & Number):
Heart 3

Distinct Symbols:
Diamond Heart Spade
```



```
Cards in Diamond Symbol:
Diamond 5
Number of Cards: 1
Sum of Numbers: 5
Average Value: 5.00
       -------------------------
Cards in Heart Symbol:
Heart 7
Heart 3
Number of Cards: 2
Sum of Numbers: 10
Average Value: 5.00
       -------------------------
Cards in Spade Symbol:
Spade 10
Spade 8
Number of Cards: 2
Sum of Numbers: 18
Average Value: 9.00
       -------------------------
```

6. **Time Complexity: O(n+klogk)**

7. **Space Complexity: O(n)**

8. **Learning Outcomes:**

   i. **Understand how to use Java classes and objects** to store structured data effectively.

   ii. **Learn the usage of `Map` (LinkedHashMap) to group and retrieve data** efficiently based on a key.

   iii. **Implement dynamic user input handling** while ensuring data validity.

   iv. **Apply sorting techniques** to display distinct keys in an organized way.