Experiment 1.1

Student Name: Ishita Verma **UID: 22BCS15761**
Branch: CSE **Section:639/A**
Semester: 6th **DOP:18/02/25**
Subject: Java **Subject Code:22CSH-359**

Aim: Write a program to collect and store all the cards to assist the users in finding all the cards in a given symbol.

Objective:

1. Collecting card details, where each card has a symbol (a character) and a number (an integer).

2. Grouping the cards based on their symbol using a Map to store the cards, where the key is the symbol (character) and the value is a list of cards (Card objects).

3. Printing out all distinct symbols in **alphabetical order** and, for each symbol:

   - Displaying the list of cards with that symbol.

   - Showing the total number of cards with that symbol.

   - Calculating and displaying the sum of the card numbers for that symbol.

Algorithm:

1. Input:

- **Prompt the user to enter the number of cards, N.**
- **For each card (from 1 to N), ask the user to input the symbol (a character) and the number (an integer) of the card.**

2. Data Structure:

- **Use a Map<Character, List<Card>> where:**
  - **The key is the symbol of the card (a char).**
  - **The value is a List<Card> which contains the cards with the same symbol.**

3. Processing:

- **For each card:**

- **Create a Card object with the provided symbol and number.**
- **Store the card in the map with its symbol as the key.**
- **If the symbol already exists in the map, add the card to the list of cards associated with that symbol.**
- **If the symbol does not exist, create a new list and add the card to it.**

4. **Sorting:**

- **Use a TreeMap instead of a regular HashMap to store the cards. This automatically sorts the keys (symbols) in alphabetical order.**

5. **Output:**

- **Iterate through the keys of the Map (which will be sorted alphabetically due to the TreeMap).**
- **For each symbol:**
  - **Print the symbol.**
  - **Print all card details associated with that symbol.**
  - **Print the total number of cards associated with that symbol.**
  - **Print the sum of the numbers on all cards with that symbol.**

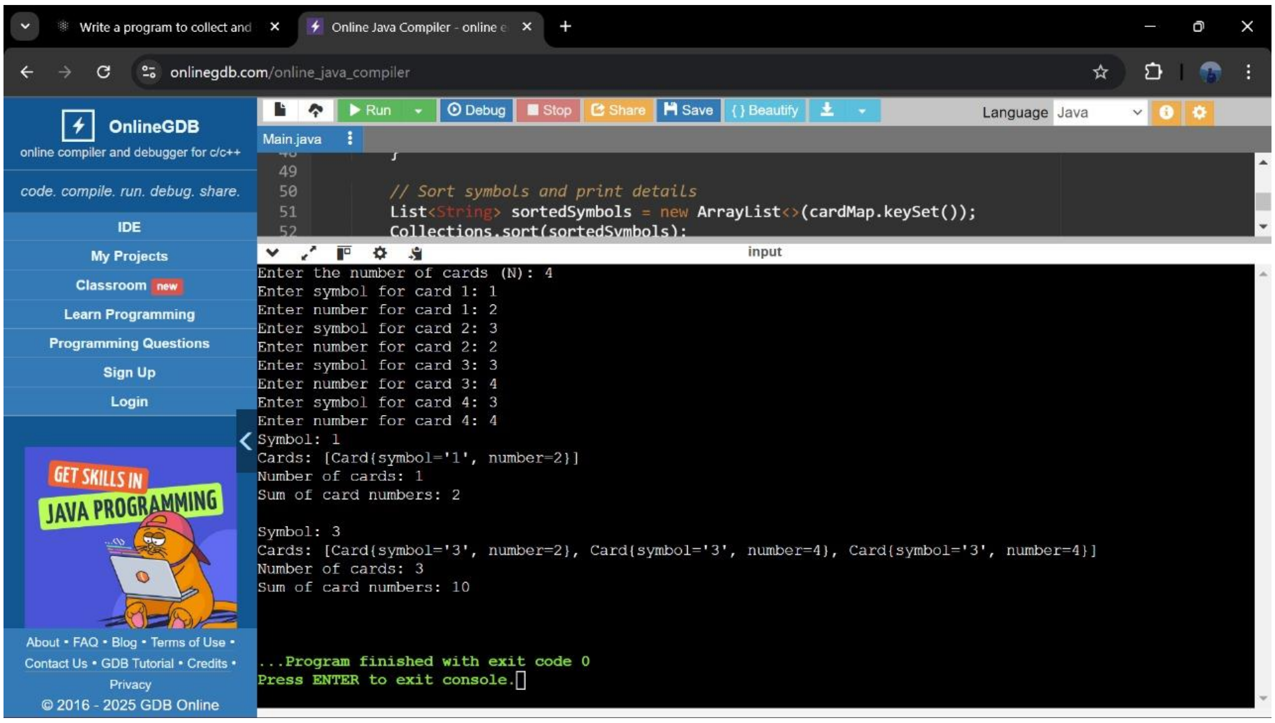
6. **End Program:**

- **Close the Scanner object to free up resources.**

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 "Card{" +
                "symbol='" + symbol + '\'' +
                ", number=" + number +
                '}';
    }
}

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

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

        for (int i = 0; i < N; i++) {
                System.out.print("Enter symbol for card " + (i + 1) + ": ");
                String symbol = scanner.nextLine();
                System.out.print("Enter number for card " + (i + 1) + ": ");
                int number = scanner.nextInt();
                scanner.nextLine(); // Consume the newline

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

        // Sort symbols and print details
        List<String> sortedSymbols = new ArrayList<>(cardMap.keySet());
        Collections.sort(sortedSymbols);

        for (String symbol : sortedSymbols) {
                List<Card> cards = cardMap.get(symbol);
                int count = cards.size();
                int sum = cards.stream().mapToInt(Card::getNumber).sum();

                System.out.println("Symbol: " + symbol);
                System.out.println("Cards: " + cards);
                System.out.println("Number of cards: " + count);
                System.out.println("Sum of card numbers: " + sum);
                System.out.println();
        }

        scanner.close();
    }
}
```

Output:


Write a program to collect and
Online Java Compiler - online e
onlinegdb.com/online_java_compiler
OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
GET SKILLS IN JAVA PROGRAMMING
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

```java
        }

        // Sort symbols and print details
        List<String> sortedSymbols = new ArrayList<>(cardMap.keySet());
        Collections.sort(sortedSymbols);
```

```
Enter the number of cards (N): 4
Enter symbol for card 1: 1
Enter number for card 1: 2
Enter symbol for card 2: 3
Enter number for card 2: 2
Enter symbol for card 3: 3
Enter number for card 3: 4
Enter symbol for card 4: 3
Enter number for card 4: 4
Symbol: 1
Cards: [Card{symbol='1', number=2}]
Number of cards: 1
Sum of card numbers: 2

Symbol: 3
Cards: [Card{symbol='3', number=2}, Card{symbol='3', number=4}, Card{symbol='3', number=4}]
Number of cards: 3
Sum of card numbers: 10


...Program finished with exit code 0
Press ENTER to exit console.
```

Learning Outcomes:

1. **Object-Oriented Programming:** Learn to create and manipulate Java classes, objects, and constructors.
2. **Collections Usage:** Understand how to use Map and List for storing and grouping data.
3. **I/O Handling:** Capture and process user input with Scanner and display formatted output.
4. **Data Grouping and Sorting:** Group data by symbols and sort keys using TreeMap.
5. **Algorithm Design:** Develop algorithms to collect, process, and display grouped data.
6. **Resource Management:** Practice proper resource management by closing the Scanner object.