**“Experiment 2.1”**

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:**

**Playing cards during travel is a fun filled experience. For this game they wanted to collect all four unique symbols. Can you help these guys to collect unique symbols from a set of cards? Create Card class with attributes symbol and number. From our main method collect each card details (symbol and number) from the user. Collect all these cards in a set, since set is used to store unique values or objects. Once we collect all four different symbols display the first occurrence of card details in alphabetical order.**

**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:**

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

Set<Card> set = new HashSet<>();

for (int i = 0; i < 8; i++) {

System.out.println("Enter a card:");

char c = in.next().charAt(0);

int n = in.nextInt();

in.nextLine();

set.add(new Card(c,n));

}

System.out.println("Four symbols gathered in eight cards.");

System.out.println("Cards in Set are:");

for (Card card : set)

System.out.println(card.symbol + " " + card.number);

in.close();

}

}

class Card implements Comparable<Card> {

char symbol;

int number;

public Card(char symbol, int number) {

this.symbol = symbol;

this.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 0;

}

@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:**

