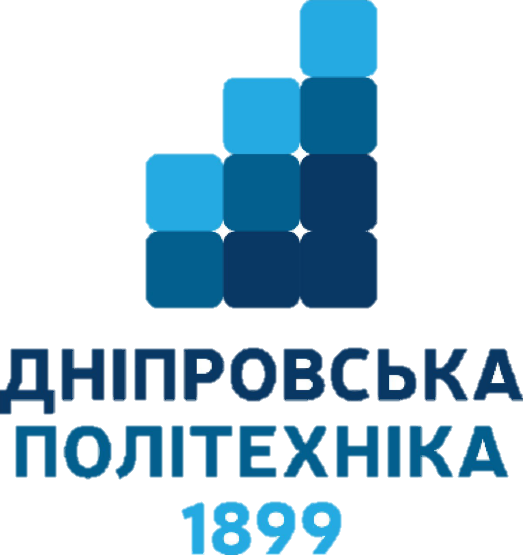
Міністерство освіти і науки України

Національний технічний університет

«Дніпровська політехніка»



Факультет інформаційних технологій

Кафедра системного аналізу і управління

*Звіт*

З дисципліни:

“ОПСJ”

**Виконав**

студент групи 124-18ск-1

Владимиров Я.Д

**GitHub**

https://github.com/gotoindex/javalabs.git

**Перевірив**

Мінєєв О.С

м. Дніпро

2020

Лабораторна робота №5

Хід роботи

Постановка задачі:

Створити гру Блек Джек (Відома як 21).  Гра карткова. Використовується  одна стандартна колода на 52 картки. Детальні правила гри можна прочитати у відкритому доступі або на Вікіпедії.

Коротко про правила гри: грає людина проти комп'ютера. Людині видається дві карти.

Мета гри набрати найбільшу кількість балів, але не більше 21. Після отримання двох карток людина має зробити вибір, або зупинити гру і залишити отриману кількість балів, або взяти ще одну картку.

Якщо кількість балів на руках перевищує 21 то людина автоматично програє. Людина може в будь-який час зупинитись. Згідно з теорією гри жодна комбінація з двох карток не може перевищувати 21 бал.

Лістинг програми:

*Run.java:*

**package** lr5;  
  
**public class** Run {  
 **public static void** main(String[] args) {  
 GameController.*run*();  
 }  
}

*GameController.java:*

**package** lr5;  
  
**import** java.util.Scanner;  
  
**public class** GameController {  
  
 **private static final** Scanner ***INPUT*** = **new** Scanner(System.***in***);  
  
 **public static void** run() {  
 Game game = **new** Game();  
 **while** (*askForNewGame*()) game.startNewGame();  
 }  
  
 **public static boolean** askForNewGame() {  
 System.***out***.print(**"Start a new game? [Y/n]: "**);  
 **return *INPUT***.hasNextLine() && (***INPUT***.nextLine().toLowerCase().equals(**"y"**));  
 }  
}

*Game.java:*

**package** lr5;  
  
**public class** Game {  
  
 **private** HumanPlayer **human**;  
 **private** ComputerPlayer **computer**;  
 **private** Deck **deck**;  
  
 **public void** startNewGame() {  
 System.***out***.println(**"--- new game has started ---"**);  
 clearAll();  
 playGame();  
 checkWhoWon();  
 }  
  
 **private void** clearAll() {  
 **this**.**human** = **new** HumanPlayer();  
 **this**.**computer** = **new** ComputerPlayer();  
 **this**.**deck** = **new** Deck();  
 **this**.**deck**.fill();  
 }  
  
 **private void** playGame() {  
 **human**.drawCard(**deck**);  
 **human**.drawCard(**deck**);  
 **computer**.makeAMove(**deck**);  
 **human**.makeAMove(**deck**);  
 }  
  
 **private void** checkWhoWon() {  
 **human**.printHand();  
 **computer**.printHand();  
 output(**human**.**hand**.getSumPoints(), **computer**.**hand**.getSumPoints());  
 }  
  
 **private void** output(**int** human\_points, **int** computer\_points) {  
 **if** (human\_points <= 21) {  
 **if** (human\_points > computer\_points || computer\_points > 21) System.***out***.println(**"You win!"**);  
 **else if** (human\_points == computer\_points) System.***out***.println(**"Draw!"**);  
 **else** System.***out***.println(**"You lose!"**);  
 }  
 **else** System.***out***.println(**"Overtake!"**);  
 }  
}

*HumanPlayer.java:*

**package** lr5;  
  
**import** java.util.Scanner;  
  
**public class** HumanPlayer **extends** Player {  
  
 **private static final** Scanner ***INPUT*** = **new** Scanner(System.***in***);  
  
 **public void** printHand() {  
 System.***out***.print(**"Your hand is: "**);  
 printAllCards();  
 }  
  
 **public void** makeAMove(Deck deck) {  
 printHand();  
 System.***out***.print(**"Do you want to draw another card? [Y/n]: "**);  
 **boolean** choice = ***INPUT***.hasNextLine() && (***INPUT***.nextLine().toLowerCase().equals(**"y"**));  
 **if** (choice) {  
 drawCard(deck);  
 **if** (**hand**.getSumPoints() <= 21) makeAMove(deck);  
 }  
 }  
}

*ComputerPlayer.java:*

**package** lr5;  
  
**public class** ComputerPlayer **extends** Player {  
  
 **public void** printHand() {  
 System.***out***.print(**"Computer's hand is: "**);  
 printAllCards();  
 }  
  
 **public void** makeAMove(Deck deck) {  
 **while** (**hand**.getSumPoints() < 17) drawCard(deck);  
 printFirstCard();  
 }  
  
 **public void** printFirstCard() {  
 System.***out***.println(**"First computer's card is: "** + **hand**.**cards**.get(0).getString());  
 }  
}

*Player.java:*

**package** lr5;  
  
**import** java.util.Random;  
  
**public abstract class** Player {  
  
 **public static final** Random ***generator*** = **new** Random();  
 **public** Deck **hand**;  
  
 **public** Player() {  
 **this**.**hand** = **new** Deck();  
 }  
  
 **public void** drawCard(Deck deck) {  
 **int** random\_card\_index = ***generator***.nextInt(deck.**cards**.size());  
 **hand**.**cards**.add(deck.**cards**.get(random\_card\_index));  
 deck.**cards**.remove(random\_card\_index);  
 }  
  
 **public void** printAllCards() {  
 **for** (Card card: **hand**.**cards**) System.***out***.print(card.getString() + **", "**);  
 System.***out***.println(**"total points: "** + **hand**.getSumPoints());  
 }  
  
 **public abstract void** printHand();  
 **public abstract void** makeAMove(Deck deck);  
}

*Deck.java:*

**package** lr5;  
  
**import** java.util.ArrayList;  
  
**public class** Deck {  
  
 ArrayList<Card> **cards**;  
  
 **public** Deck() {  
 **this**.**cards** = **new** ArrayList<>();  
 }  
  
 **public int** getSumPoints() {  
 **int** sum = 0;  
 **for** (Card card: **cards**) sum += card.**value**.getPoints();  
 **return** sum;  
 }  
  
 **public void** fill() {  
 **for** (CardSuit suit: CardSuit.*values*()) {  
 **for** (CardValue value: CardValue.*values*()) **cards**.add(**new** Card(value, suit));  
 }  
 }  
}

*Card.java:*

**package** lr5;  
  
**public class** Card {  
  
 **public** CardValue **value**;  
 **public** CardSuit **suit**;  
  
 **public** Card(CardValue value, CardSuit suit) {  
 **this**.**value** = value;  
 **this**.**suit** = suit;  
 }  
  
 **public** String getString() {  
 **return value**.getName() + **suit**;  
 }  
}

*CardSuit.java:*

**package** lr5;  
  
**public enum** CardSuit {  
 ***D***, *// Diamonds* ***C***, *// Clubs* ***H***, *// Hearts* ***S*** *// Spades*}

*CardValue.java:*

**package** lr5;  
  
**public enum** CardValue {  
 ***TWO***(**"2"**, 2), *// Two* ***THREE***(**"3"**, 3), *// Three* ***FOUR***(**"4"**, 4), *// Four* ***FIVE***(**"5"**, 5), *// Five* ***SIX***(**"6"**, 6), *// Six* ***SEVEN***(**"7"**, 7), *// Seven* ***EIGHT***(**"8"**, 8), *// Eight* ***NINE***(**"9"**, 9), *// Nine* ***TEN***(**"10"**, 10), *// Ten* ***JACK***(**"J"**, 10), *// Jack* ***QUEEN***(**"Q"**, 10), *// Queen* ***KING***(**"K"**, 10), *// King* ***ACE***(**"A"**, 1); *// Ace* **private final** String **name**;  
 **private final int points**;  
  
 CardValue(String s, **int** i) {  
 **this**.**name** = s;  
 **this**.**points** = i;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public int** getPoints() {  
 **return points**;  
 }  
}

Результати виконання:

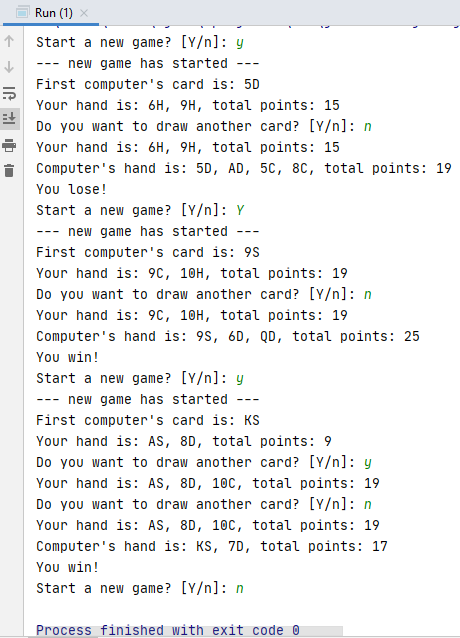


Рисунок 5.1 – Результат виконання програми

Висновок: Під час виконання лабораторної роботи №5 було створено гру Блек Джек.