**자료구조(박요한 교수님)**

**동일한 카드를 뽑을 확률 구하기 과제**

컴퓨터공학과 유상현

**1. 프로그램 소스코드**

**App.java**

|  |
| --- |
| package Day9\_2.App;  import java.util.\*;  import Day9\_2.lib.Card;  class OneCard extends Card{  enum Patten {  clover("클로버"), //0은 클로버  heart("하트"), //1은 하트  Diamond("다이아몬드"), //2는 다이야  Space("스페이스"); //3은 스페이  public String korName;  Patten(String name){ this.korName = name; }  String GetName(){ return this.korName; }  }  enum Number{  ONE("1"), TWO("2"), THREE("3"), FOUR("4"), FIVE("5"),  SIX("6"), SEVEN("7"), EIGHT("8"), NINE("9"), TEN("10"),  JACK("J"), QUEEN("Q"), KING("K"), ACE("A");  private String symbol;  Number(String symbol) { this.symbol = symbol; }  String GetNum(){ return this.symbol; }  }  @Override  public void draw() {  Random random = new Random();  int patten = random.nextInt(4);  int number = random.nextInt(13);  setData(patten, number);  }  @Override  public void showCard() {  System.out.printf("OneCard: %s %s", Patten.values()[this.GetPatten()].GetName(), Number.values()[this.GetNumber()].GetNum());  }  @Override  public int compareTo(Card o) {  if (this.GetPatten() > o.GetPatten()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Patten.values()[this.GetPatten()],  Number.values()[this.GetNumber()].GetNum());  return 1;  } else if (this.GetPatten() < o.GetPatten()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Patten.values()[o.GetPatten()],  Number.values()[o.GetNumber()].GetNum());  return -1;  } else {  if (this.GetNumber() > o.GetNumber()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Patten.values()[this.GetPatten()],  Number.values()[this.GetNumber()].GetNum());  return 1;  } else if (this.GetNumber() < o.GetNumber()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Patten.values()[o.GetPatten()],  Number.values()[o.GetNumber()].GetNum());  return -1;  } else {  return 0;  }  }  }  public String toString(){  return "OneCard:" + Patten.values()[this.GetPatten()].GetName() + Number.values()[this.GetNumber()].GetNum();  }  public boolean equals(OneCard onecard){  if(this.GetNumber() == onecard.GetNumber() && this.GetPatten() == onecard.GetPatten()){  return true;  }  else{  return false;  }  }  }  class UnoCard extends Card{  //enum  public enum UnoNum {  One("1"), Two("2"), Three("3"), Four("4"), Five("5"),  Six("6"), Seven("7"), Eight("8"), Nine("9"), Ten("10"),  Skip("Skip"), Reverse("Reverse"), Draw("Draw");    private String RealNumber;  UnoNum(String RealNumber) {this.RealNumber = RealNumber;}  String GetRealNumber() {return this.RealNumber;}  }  public enum Color {  Red, Yellow, Green, Blue;  }  public String toString(){  return "UnoCard:"+ Color.values()[this.GetPatten()] + " " + UnoNum.values()[this.GetNumber()].GetRealNumber();  }  @Override  public void draw() {  Random random = new Random();  int patten = random.nextInt(4);  int number = random.nextInt(13);  setData(patten, number);  }  @Override  public void showCard() {  System.out.printf("UnoCard: %s %s", Color.values()[this.GetPatten()],  UnoNum.values()[this.GetNumber()].GetRealNumber());  }  @Override  public int compareTo(Card o) {  if (this.GetPatten() > o.GetPatten()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Color.values()[this.GetPatten()],  UnoNum.values()[this.GetNumber()].GetRealNumber());  return 1;  } else if (this.GetPatten() < o.GetPatten()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Color.values()[o.GetPatten()],  UnoNum.values()[o.GetNumber()].GetRealNumber());  return -1;  } else {  if (this.GetNumber() > o.GetNumber()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Color.values()[this.GetPatten()],  UnoNum.values()[this.GetNumber()].GetRealNumber());  return 1;  } else if (this.GetNumber() < o.GetNumber()) {  System.out.printf("더 큰 카드는 %s %s 입니다.\n", Color.values()[o.GetPatten()],  UnoNum.values()[o.GetNumber()].GetRealNumber());  return -1;  } else {  return 0;  }  }  }  }  public class App {  static int sameCard(OneCard card, double input){  OneCard onecard = new OneCard();  onecard.draw();  boolean IsSame = onecard.equals(card);  int count = 0;  for(int i = 0; i < input; i++){  if(IsSame){  count++;  }  onecard.draw();  IsSame = onecard.equals(card);  }  double result = count/input;  System.out.println("동일한 카드가 뽑힐 확률: "+ result);  return count;  }  public static void main(String[] args) throws Exception {  OneCard p1 = new OneCard();  OneCard p2 = new OneCard();  UnoCard up1 = new UnoCard();  p1.draw();  p2.draw();  up1.draw();  System.out.println(p1);  System.out.println(p2);  System.out.println(up1);  sameCard(p1, 1000);  }  } |

**Card.java**

|  |
| --- |
| package Day9\_2.lib;  public abstract class Card implements Comparable<Card>{  private int patten, number;  public Card(){};  public abstract void draw(); // 카드 뽑는 기능  public abstract void showCard();  public abstract int compareTo(Card o);  public void setData(int patten, int number) {  this.patten = patten;  this.number = number;  }  public int GetPatten(){  return this.patten;  }  public int GetNumber(){  return this.number;  }  } |

**2. 실행 화면**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**3. 느낀점(고찰)**

이번 실습 과제를 통해서 자바의 클래스 파일을 분리해서 관리하는 방법인 패키지를 이해할 수 있었습니다. 그리고 자바의 내장된 패키지가 굉장히 많다는 것을 알게 되었습니다. 그리고 toString 이라는 패키지를 통해서 그냥 클래스의 값을 찍었을 때 클래스의 주소 값이 아니라 내가 지정한 값이 나오게 만들 수 있다는 것을 알게 되었습니다. 아직은 어디에 어떻게 쓰이는지 잘 알 수는 없었지만 클래스의 값을 잘못 호출하는 경우를 막을 수 있겠다는 생각은 들었습니다.