**자료구조(박요한 교수님)**

**개별 카드내용 비교 가능한 클래스 만들기 과제**

컴퓨터공학과 유상현

**1. 프로그램 소스코드**

**App.java**

|  |
| --- |
| import java.util.\*;  abstract class Card{  private int patten, number;  public Card(){};  public abstract void draw(); // 카드 뽑는 기능  public abstract void showCard();  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;  }  }  class OneCard extends Card implements Comparable<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;  }  }  }  }  class UnoCard extends Card implements Comparable<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;  }  @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", 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 {  public static void main(String[] args) throws Exception {  OneCard p1 = new OneCard();  OneCard p2 = new OneCard();  UnoCard up1 = new UnoCard();  UnoCard up2 = new UnoCard();  p1.draw();  p2.draw();  p1.compareTo(p2);  up1.draw();  up2.draw();  up1.compareTo(up2);  }  } |

**2. 실행 화면**

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

**3. 느낀점(고찰)**

이번 실습 과제를 통해서 추상 클래스와 인터페이스에 대해서 알게 되었습니다. 그리고 자바의 인터페이스에는 내장 인터페이스가 있다는 것을 알게 되었습니다. 처음에는 그냥 부모클래스와 추상클래스의 사용 차이에 대해서 잘 이해하지 못했는데 추상클래스를 이용하면 처음에 프로그램을 짜기 전에 틀을 잘 세울 수 있을 것 같다는 생각이 들었습니다.