Programación Orientada a Objetos

Parcial I

1. Se desea desarrollar un programa que permita jugar 21. Para esto se debe diseñar una clase que represente una carta. Una Carta corresponde a un valor (AS,2,3,4,5,6,7,8,9,10,J,Q,K) y a un palo (Diamantes, Corazones, Picas y Tréboles). La clase debe proveer los siguientes métodos:

getValorReal(): devuelve el valor real de la carta. En el juego de la 21, las cartas J, Q y K valen 10. El AS vale 1 u 11. Para este ejercicio asumir que siempre vale 1.

toString(): devuelve un string con la información de la carta, por ej: 2 de tréboles

valeMasQue(): Este método compara 2 cartas. Devuelve -1 si la primera carta vale más que la segunda, 0 si valen igual, y 1 si la segunda vale más que la primera.

Crear un programa de prueba con las siguientes cartas

C1 = 2 de picas

C2 = 4 de diamantes

C3 = J de tréboles

C4 = k de corazones

Imprimir(C1 +”vale “+valor\_de\_c1)

Imprimir(C4 +”vale “+valor\_de\_c4)

Si(c1 vale mas que c2)

Imprmir c1 + “ vale mas que “ +c2

Si no si Imprmir c2 + “ vale mas que “ +c1

Imprmir c1 + “ vale mas que “ +c2

Si no

Imprmir c1 + “es igual a “ +c2

Calificación:

**Construcción de la clase: 15 puntos**

**Programa de Prueba: 5 puntos**

1. Escribir una clase que permita representar un polinomio como máximo de grado 4.

La clase debe proveer los siguientes métodos:

Sumar: Permite la suma de 2 polinomios

Derivar: permite encontrar la derivada de un polinomio

Evaluar: evalúa el polinomio para un x dado

toString: devuelve un string con la representación del polinomio

ej: 2x^2 + 3

Realizar un programa de prueba:

P1 = 3x^4 + 2x^2 +1

P2 = 2x^2 + 3

P3 = derivada de P1

P4 = p3 + p2

y = p4 evaluado en 3

Imprimir(P1)

Imprimir(P2)

Imprimir(P3)

Imprimir(P4)

Imprimir(p4 + “evaluado en 3 es “ + y)

**Construcción de la clase: 25 puntos**

**Programa de Prueba: 5 puntos**