UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGIAS
Carrera de Ingeniería Informática

Materia: Estructura de Datos I

INF 220

"Tema 2. TAD Polinomios"

Docente: Ing. Ubaldo Pérez Ferreira
E-mail: ubaperez@gmail.com

Especificación Formal – TAD Polinomios

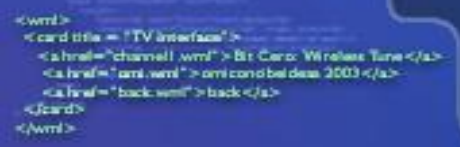
$P(x) = 5x^3 + 3x^2 – 2x + 2$

Operaciones

Evaluar $P(2) = 50$

Operaciones Binarias: C, P, Q $\in$ Polinomios

$C(x) = P(X) + Q(X)$



Especificación Formal – TAD Polinomios

Podemos Especificar un TAD para los Polinomios

Structure Polinomio

F

Declare Zero() → Poly	//Define polinomio
Iszero(Poly) → Boolean	//Esta vació el Polinomio
Coef(Poly, exp) → coef	//Obtiene el coef. del Polinomio
Attach(Poly, coef, exp) → Poly	//Adiciona un elemento al Polinomio
Rem(Poly, exp) → Poly	//Elimina un elemento del Polinomio
Smult(Poly, coef, exp)→ Poly	// Multiplicación por un monomio
Add(Poly, Poly)→ Poly	//Adición de Polinomios
Mult(Poly, Poly) → Poly	//Multiplicación de Polinomios

Especificación Formal – TAD Polinomios

$\forall$ P,Q $\in$ Poly c , d $\in$ coef, e,f $\in$ exp

A

Rem(Zero, f)::= Zero

Rem(Attach(P, c, e),f)::= If e=f Then Rem(P,f) Else Attach(Rem(P,f),c,e)

Iszero(Zero)::=true

Iszero(Attach(P,c,e))::= If Coef(P,e)=-c then Iszero(Rem(P,e)) else false

Coef(Zero,e)::=0

Coef(Attach(P,c,e),f)::= If e=f then c + Coef(P,f) else Coef(P,f)

Smult(Zero,d,f)::=Zero

Smult(Attach(P,c,e),d,f)::= Attach(Smult(P,d,f), cxd, e+f)

Add(P, Zero)::=P

Add(P,Attach(Q,d,f))::=Attach(Add(P,Q),d,f)

Mult(P,Zero)::=Zero

Mult(P,Attach(Q,d,f))::= Add(Mult(P,Q),Smult(P,d,f))

Ejercicio de Especificación Formal – TAD Polinomios

Dado el Polinomio P y Q, diseñar el algoritmo para sumar dos polinomios (Add), utilizando las Funciones y Axiomas del TAD Polinomio. Se debe considerar la función Grado(Poly) la misma que obtiene el grado de un polinomio o exponente mayor.

```
Procedurre Add()
C= Zero
While NOT (IsZero(P) and IsZero(Q)) do
        Case
        :Grado(P)<Grado (Q)
                C=Attach(C, Coef(Q, Grado(Q)), Grado(Q))
                Q=Rem(Q, Grado(Q))
        :Grado(P)>Grado (Q)
                C=Attach(C,Coef(P, Grado(P)), Grado(P))
                P=Rem(P, Grado(P))
        :Grado(P)=Grado (Q)
                C=Attach(C,Coef(P, Grado(P))+ Coef(Q, Grado(Q)), Grado(P))
                P=Rem(P, Grado(P))
                Q=Rem(Q, Grado(Q))
        End Case
End while
```

Ejercicio de Especificación Formal – TAD Polinomios

Dado el Polinomio P(x)= $3x^6+2x^4+3$ y el polinomio Q(x)=3x-1 cual es el resultado de aplicar las siguientes expresiones utilizando las Funciones y Axiomas del TAD Polinomio

Rem(P, 4)

Rem(Attach(P,-8,5,),0)

Iszero(Attach(P,0,0))

Smult(Zero,3,6)

Smult(Attach(P,2,2),1,1)

Attach(Smult(P,1,1), 1*2, 2+1)

Add(P, Zero)

Add(P,Attach(P,3,2))

Attach(Add(P,Q),3,2)

Mult(P,Zero)

Mult(P,Attach(Q,4-2,3))

Add(Mult(P,Q),Smult(P,4-2,3)

Ejercicio Implementacion – TAD Polinomios

```
public class clsPoly
{
    const int MAX = 100;
    float[] vCoef;
    int[] vExp;
    int nTerm;

    public clsPoly()
    {
        vCoef = new float[MAX];
        vExp = new int[MAX];
        for (int i = 0; i < MAX; i++)
             {
                vCoef[i] = 0;
                vExp[i] = 0;
             }
        nTerm = 0;
    }

    public clsPoly(clsPoly P)
    {
        vCoef = new float[MAX];
        vExp = new int[MAX];
        for (int i = 0; i <= P.nTerm; i++)
        {
            vCoef[i] = P.vCoef[i];
            vExp[i] = P.vExp[i];
        }
        this.nTerm = P.nTerm;

    }
```

```
    // Declare el Polinomio Zero
    public clsPoly Zero()
    {
        return new clsPoly();
    }

    // Es Zero un Polinomio P
    public bool IsZero()
    {
        return (nTerm == 0);
    }

    // Adciona un termino al polinomio
    public clsPoly Attach(clsPoly P, float Coef, int Exp)
    {
        if ((Coef != 0) && (Exp >= 0))
        {
            P.nTerm++;
            P.vExp[Exp] = Exp;
            P.vCoef[Exp] = P.vCoef[Exp] + Coef;
        }
        return P;
    }
```

Ejercicio Implementacion – TAD Polinomios

```
// Elimina el termino con exponente Exp del polinomio P
public clsPoly Rem(clsPoly P, int Exp)
{

    if (Exp >= 0)
    {
        P.vCoef[Exp] = 0;
        P.vExp[Exp] = 0;
        P.nTerm--;
    }
    return P;
}


// Obtiene el coheficiente de un termino con exponente Exp del polinomio
public float Coef( int Exp)
{
        return vCoef[Exp];
}

// Obtiene el grado de un Polinomio
public int Grado()
{
    int Exp=0;
    for (int k = 0; k < MAX; k++)
    {
        if (vExp[k]>0)
            Exp=vExp[k];
    }
    return Exp;
}
```

Ejercicio Implementacion – TAD Polinomios

```
// Suma de dos polinomios
public clsPoly Add(clsPoly P, clsPoly Q)
{
    clsPoly C = new clsPoly();

    while ((P.IsZero()==true && Q.IsZero()==true)==false)
    {
        if (P.Grado() < Q.Grado())
            {
                C = Attach(C, Q.Coef(Q.Grado()), Q.Grado());
                Q = Rem(Q, Q.Grado());
            }
        if (P.Grado() > Q.Grado())
            {
                C = Attach(C, P.Coef(P.Grado()), P.Grado());
                P = Rem(P, P.Grado());

            }
        if (P.Grado() == Q.Grado())
            {
                C = Attach(C, P.Coef( P.Grado()) + Q.Coef( Q.Grado()), P.Grado());
                P = Rem(P, P.Grado());
                Q = Rem(Q, Q.Grado());
            }
    }
    return C;
}
```