

TDA Polinomio

JotaEle Díaz y Gregorio Vidoy

1.0

Dic 2014

Índice de clases

Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

- **polinomio (TDA polinomio)**
- **polinomio::const_iterator (Class const_iterator Iterador constante hacia delante sobre todas las posiciones del vectorD. Lectura const_iterator, const_iterator(const_iterator), const_iterator(iterator), operator*, operator++, ++operator, operator=(iterator), operator==, operator!=)**
- **polinomio::iterator (Class iterator Iterador hacia delante sobre todas las posiciones del polinomio itera sobre las oposiciones != nulo. iterator, operator*, operator++, ++operator operator=(iterator), operator==, operator!=)**

Referencia de la Clase polinomio

TDA que permite la representación de polinomios. Permite realizar varias operaciones con polinomios como la suma y resta de ellos. polinomios suma y resta de polinomios.

Gracias a la implementación interna usando un Vector Disperso (vectorD), el recorrido y acceso sobre este es eficiente y su almacenamiento ocupa una menor cantidad de memoria.

La gestión de memoria es dinámica y el tamaño del polinomio es modificable a medida que se van añadiendo monomios a éste.

```
#include <polinomio.h>
```

Clases

- **class const_iterator**
*Iterador constante hacia delante sobre todas las posiciones del vectorD. Lectura **const_iterator**, **const_iterator(const_iterator)**, **const_iterator(iterator)**, **operator***, **operator++**, **++operator**, **operator=(iterator)**, **operator==**, **operator!=** class **iterator***
- **class iterator**
*Iterador hacia delante sobre todas las posiciones del polinomio itera sobre las oposiciones != nulo. **iterator**, **operator***, **operator++**, **++operator** **operator=(iterator)**, **operator==**, **operator!=***

Métodos públicos

- **polinomio ()**
Constructor por defecto.
- **polinomio (const polinomio &otro)**
Constructor primitivo que hace una copia de un polinomio.
- **~polinomio ()**
Destructor primitivo de vectorD.
- **int getGrado ()**
Consulta el grado del polinomio.
- **int getCoficiente (int gm)**
Consulta el coeficiente de un monomio.
- **void setMonomio (int c, int g)**
Añade un nuevo monomio al polinomio de coeficiente c y grado g (cx^g)
- **polinomio & operator= (const polinomio &otro)**
Operador de asignación. Asigna una copia del polinomio al recibido como argumento.
- **polinomio operator+ (const polinomio &otro)**
Operador suma. Suma al polinomio el polinomio recibido como argumento.
- **polinomio operator- (polinomio &otro)**
Operador diferencia. Resta al polinomio el polinomio recibido como argumento.
- **polinomio operator- ()**
Operador unario -. Da como resultado un polinomio que es igual al polinomio original en negativo (-polinomio).
- **iterator begin ()**
Devuelve un iterador que apunta al primer elemento del polinomio.
- **iterator end ()**
Devuelve un iterador que apunta al final (es decir, al elemento siguiente al último) del polinomio.
- **const_iterator cbegin () const**
Devuelve un iterador constante que apunta al primer elemento del polinomio.
- **const_iterator cend () const**
Devuelve un iterador constante que apunta al final (es decir, al elemento siguiente al último) del polinomio.

Amigas

- **ostream & operator<< (ostream &os, const polinomio &p)**
Imprime por la salida estándar un polinomio.

Documentación del constructor y destructor

polinomio::polinomio (const polinomio & otro)

Constructor primitivo que hace una copia de un polinomio.

Parámetros:

in	otro	Polinomio que se desea copiar
----	------	-------------------------------

Documentación de las funciones miembro

- **polinomio::iterator polinomio::begin ()**

Devuelve un iterador que apunta al primer elemento del polinomio.

Devuelve:

Iterador a la primera posición.

Postcondición:

No se modifica el polinomio.

- **polinomio::const_iterator polinomio::cbegin () const**

Devuelve un iterador constante que apunta al primer elemento del polinomio.

Devuelve:

Iterador constante a la primera posición.

Postcondición:

No se modifica el polinomio.

- **polinomio::const_iterator polinomio::cend () const**

Devuelve un iterador constante que apunta al final (es decir, al elemento siguiente al último) del polinomio.

Devuelve:

Iterador constante al elemento siguiente al último.

Postcondición:

No se modifica el polinomio.

- **polinomio::iterator polinomio::end ()**

Devuelve un iterador que apunta al final (es decir, al elemento siguiente al último) del polinomio.

Devuelve:

Iterador al elemento siguiente al último.

Postcondición:

No se modifica el polinomio.

- **int polinomio::getCoeficiente (int *gm*)**

Consulta el coeficiente de un monomio.

Parámetros:

in	<i>gm</i>	grado del monomio
----	-----------	-------------------

Devuelve:

retorna su coeficiente (int)

- **int polinomio::getGrado ()**

Consulta el grado del polinomio.

Devuelve:

el grado del polinomio (El mayor de los grados de los monomios que lo componen) (int)

- **polinomio polinomio::operator+ (const polinomio & *otro*)**

Operador suma. Suma al polinomio el polinomio recibido como argumento.

Parámetros:

in	<i>otro</i>	polinomio a sumar
----	-------------	-------------------

Devuelve:

polinomio suma de ambos

Postcondición:

El polinomio original no se ve alterado.

- **polinomio polinomio::operator- (polinomio & *otro*)**

Operador diferencia. Resta al polinomio el polinomio recibido como argumento.

Parámetros:

in	<i>otro</i>	polinomio a restar
----	-------------	--------------------

Devuelve:

Polinomio diferencia de ambos

Postcondición:

El polinomio original no se ve alterado.

- **polinomio polinomio::operator- ()**

Operador unario -. Da como resultado un polinomio que es igual al polinomio original en negativo (-polinomio).

Devuelve:

Polinomio en negativo (-polinomio).

Postcondición:

El polinomio original no se ve alterado.

- **polinomio & polinomio::operator= (const polinomio & otro)**

Operador de asignación. Asigna una copia del polinomio al recibido como argumento.

Parámetros:

in	<i>otro</i>	polinomio a copiar
----	-------------	--------------------

Devuelve:

polinomio copiado

Postcondición:

Los valores los valores del polinomio quedan igual que otro polinomio.

- **void polinomio::setMonomio (int c, int g)**

Añade un nuevo monomio al polinomio de coeficiente c y grado g (cx^g)

Parámetros:

in	<i>c</i>	Coeficiente del nuevo monomio
in	<i>g</i>	Grado del nuevo monomio

Postcondición:

Si ya existía en el polinomio un monomio con el mismo grado, se sustituirá por el nuevo.

Documentación de las funciones relacionadas y clases amigas

- **ostream& operator<< (ostream & os, const polinomio & p) [friend]**

Imprime por la salida estándar un polinomio.

Parámetros:

in	<i>os</i>	Salida del polinomio
in	<i>p</i>	Polinomio a imprimir

La documentación para esta clase fue generada a partir de los siguientes ficheros:

polinomio.h
polinomio.hxx

Referencia de la Clase polinomio::iterator

class iterator Iterador hacia delante sobre todas las posiciones del polinomio itera sobre las oposiciones != nulo. iterator, operator*, operator++, ++operator operator=(iterator), operator==, operator!=

```
#include <polinomio.h>
```

Métodos públicos

- **iterator ()**
Constructor primitivo del iterador sobre el VectorD.
- **iterator (const iterator &d)**
Constructor primitivo que crea un iterador sobre un VectorD copia de otro iterador.
- **const pair< int, int > &operator* ()**
Operador de indirección. Devuelve el elemento del VectorD al cual hace referencia el iterador.
- **iterator &operator++ ()**
Operador de incremento (++it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **iterator operator++ (int)**
Operador de incremento (it++). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **iterator &operator-- ()**
Operador de decremento (--it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **iterator operator-- (int)**
Operador de decremento (it--). Decrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **bool operator== (const iterator &d)**
Operador de igualdad. Comprueba si dos iteradores hacen referencia a una misma posición del polinomio.
- **bool operator!= (const iterator &d)**
Operador de desigualdad. Comprueba si dos iteradores hacen referencia a distintas posiciones del VectorD.
- **iterator &operator= (const iterator &d)**
Operador de asignación. Asigna al iterador la posición a la que apunta el iterador recibido.

Amigas

class polinomio

Descripción detallada

class iterator Iterador hacia delante sobre todas las posiciones del polinomio itera sobre las oposiciones != nulo. iterator, operator*, operator++, ++operator operator=(iterator), operator==, operator!=

Documentación del constructor y destructor

- **polinomio::iterator::iterator (const iterator & d)**

Constructor primitivo que crea un iterador sobre un VectorD copia de otro iterador.

Parámetros:

in	d	Iterador al vector disperso que se desea copiar
----	---	---

Documentación de las funciones miembro

- **bool polinomio::iterator::operator!= (const iterator & d)**

Operador de desigualdad. Comprueba si dos iteradores hacen referencia a distintas posiciones del VectorD.

Parámetros:

in	d	Iterador a comparar
----	---	---------------------

Devuelve:

True si hacen referencia distintas posiciones, false en caso contrario

- **const pair< int, int > & polinomio::iterator::operator* ()**

Operador de indirección. Devuelve el elemento del VectorD al cual hace referencia el iterador.

Devuelve:

la referencia a un pair <coeficiente,grado>.

- **polinomio::iterator & polinomio::iterator::operator++ ()**

Operador de incremento (++it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador a la posición siguiente a la actual

Postcondición:

Se modifica el iterador.

- **polinomio::iterator polinomio::iterator::operator++ (int)**

Operador de incremento (it++). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador original antes de ser incrementado.

- **polinomio::iterator & polinomio::iterator::operator-- ()**

Operador de decremento (–it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador a la posición anterior a la actual.

Postcondición:

Se modifica el iterador.

- **polinomio::iterator polinomio::iterator::operator-- (int)**

Operador de decremento (it–). Decrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador original antes de ser decrementado.

- **polinomio::iterator & polinomio::iterator::operator= (const iterator & d)**

Operador de asignación. Asigna al iterador la posición a la que apunta el iterador recibido.

Parámetros:

in	<i>d</i>	Iterador a copiar.
----	----------	--------------------

- **bool polinomio::iterator::operator== (const iterator & d)**

Operador de igualdad. Comprueba si dos iteradores hacen referencia a una misma posición del polinomio.

Parámetros:

in	<i>d</i>	Iterador a copiar.
----	----------	--------------------

Devuelve:

True si hacen referencia a la misma posición, false en caso contrario.

La documentación para esta clase fue generada a partir de los siguientes ficheros:

polinomio.h
polinomio.hxx

Referencia de la Clase polinomio::const_iterator

class **const_iterator** Iterador constante hacia delante sobre todas las posiciones del vectorD. Lectura **const_iterator**, const_iterator(const_iterator), const_iterator(iterator), operator*, operator++, ++operator, operator=(iterator), operator==, operator!=

```
#include <polinomio.h>
```

Métodos públicos

- **const_iterator ()**
Constructor primitivo del iterador sobre el VectorD.
- **const_iterator (const const_iterator &d)**
Constructor primitivo que crea un iterador constante sobre un VectorD copia de otro iterador constante.
- **const pair< int, int > &operator* ()**
Operador de indirección. Devuelve el elemento del VectorD al cual hace referencia el iterador.
- **const_iterator &operator++ ()**
Operador de decremento (++it). Incrementa el iterador constante, es decir referencia el iterador a la posición siguiente a la actual.
- **const_iterator operator++ (int)**
Operador de decremento (it++). Devuelve un iterador constante a la posición siguiente a la referenciada por el iterador actual.
- **const_iterator &operator-- ()**
Operador de decremento (--it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **const_iterator operator-- (int)**
Operador de decremento (it--). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.
- **bool operator== (const const_iterator &d)**
Operador de igualdad. Comprueba si dos iteradores hacen referencia a una misma posición del VectorD.
- **bool operator!= (const const_iterator &d)**
Operador de desigualdad. Comprueba si dos iteradores hacen referencia a distintas posiciones del VectorD.
- **const_iterator &operator= (const const_iterator &d)**
Operador de asignación. Asigna al iterador la posición a la que apunta el iterador recibido.

Amigas

class **polinomio**

Descripción detallada

class **const_iterator** Iterador constante hacia delante sobre todas las posiciones del vectorD. Lectura **const_iterator**, const_iterator(const_iterator), const_iterator(iterator), operator*, operator++, ++operator, operator=(iterator), operator==, operator!=

Documentación del constructor y destructor

- **polinomio::const_iterator::const_iterator (const const_iterator & d)**

Constructor primitivo que crea un iterador constante sobre un VectorD copia de otro iterador constante.

Parámetros:

in	<i>d</i>	Iterador al vector disperso que se desea copiar
----	----------	---

Documentación de las funciones miembro

- **bool polinomio::const_iterator::operator!= (const const_iterator & d)**

Operador de desigualdad. Comprueba si dos iteradores hacen referencia a distintas posiciones del VectorD.

Parámetros:

in	<i>d</i>	Iterador a comparar.
----	----------	----------------------

Devuelve:

True si hacen referencia distintas posiciones, false en caso contrario

- **const pair< int, int > & polinomio::const_iterator::operator* ()**

Operador de indirección. Devuelve el elemento del VectorD al cual hace referencia el iterador.

Devuelve:

la referencia a un pair <coeficiente,grado>.

- **polinomio::const_iterator & polinomio::const_iterator::operator++ ()**

Operador de decremento (++it). Incrementa el iterador constante, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador constante a la posición siguiente a la actual.

- **polinomio::const_iterator polinomio::const_iterator::operator++ (int)**

Operador de decremento (it++). Devuelve un iterador constante a la posición siguiente a la referenciada por el iterador actual.

Devuelve:

Iterador constante original antes de ser incrementado.

- **polinomio::const_iterator & polinomio::const_iterator::operator-- ()**

Operador de decremento (–it). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador constante a la posición siguiente a la actual.

- **polinomio::const_iterator polinomio::const_iterator::operator-- (int)**

Operador de decremento (it–). Incrementa el iterador, es decir referencia el iterador a la posición siguiente a la actual.

Devuelve:

Iterador original antes de ser decrementado.

- **polinomio::const_iterator & polinomio::const_iterator::operator= (const const_iterator & d)**

Operador de asignación. Asigna al iterador la posición a la que apunta el iterador recibido.

Parámetros:

in	<i>d</i>	Iterador a copiar.
----	----------	--------------------

- **bool polinomio::const_iterator::operator== (const const_iterator & d)**

Operador de igualdad. Comprueba si dos iteradores hacen referencia a una misma posición del VectorD.

Parámetros:

in	<i>d</i>	Iterador a comparar.
----	----------	----------------------

Devuelve:

True si hacen referencia a la misma posición, false en caso contrario

La documentación para esta clase fue generada a partir de los siguientes ficheros:

polinomio.h
polinomio.hxx