

Proyecto Individual

1.1

Generado por Doxygen 1.8.6

Jueves, 8 de Octubre de 2015 23:06:02

Índice general

1	Índice de clases	1
1.1	Lista de clases	1
2	Documentación de las clases	3
2.1	Referencia de la Clase ConvexHull	3
2.2	Referencia de la Clase HE_edge	3
2.2.1	Descripción detallada	4
2.2.2	Documentación del constructor y destructor	4
2.2.2.1	HE_edge	4
2.2.2.2	HE_edge	4
2.3	Referencia de la Clase HE_face	4
2.3.1	Descripción detallada	5
2.3.2	Documentación del constructor y destructor	5
2.3.2.1	HE_face	5
2.3.2.2	HE_face	5
2.3.3	Documentación de los datos miembro	5
2.3.3.1	edge	5
2.4	Referencia de la Clase HE_vert	5
2.4.1	Descripción detallada	6
2.4.2	Documentación del constructor y destructor	6
2.4.2.1	HE_vert	6
2.4.3	Documentación de los datos miembro	6
2.4.3.1	edge	6
2.4.3.2	point	6
2.5	Referencia de la Clase Point2D	6
2.5.1	Descripción detallada	7
2.5.2	Documentación del constructor y destructor	7
2.5.2.1	Point2D	7
2.5.2.2	Point2D	7
2.5.2.3	Point2D	7
2.5.3	Documentación de las funciones miembro	8

2.5.3.1	assignn	8
2.5.3.2	get	8
2.5.3.3	getAm	8
2.5.3.4	getLeftMost	8
2.5.3.5	getx	8
2.5.3.6	gety	8
2.5.3.7	onLeft	8
2.5.3.8	operator!=	8
2.5.3.9	operator-	8
2.5.3.10	operator==	9
2.5.3.11	setNext	9
2.5.3.12	setNext	9
2.5.3.13	sett	9
2.5.3.14	setx	9
2.5.3.15	sety	9

Índice**10**

Capítulo 1

Índice de clases

1.1. Lista de clases

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

ConvexHull	3
HE_edge	
La clase HE_edge	3
HE_face	
La clase HE_face	4
HE_vert	
La clase HE_vert	5
Point2D	
Clase Point2D	6

Capítulo 2

Documentación de las clases

2.1. Referencia de la Clase ConvexHull

Métodos públicos

- **ConvexHull** ([Point2D](#))
- **ConvexHull** ([Point2D](#), [HE_face](#) *)
- **ConvexHull** (const [ConvexHull](#) &obj)

Atributos públicos

- [HE_face](#) * **convexhull**
- [Point2D](#) **points**

La documentación para esta clase fue generada a partir del siguiente fichero:

- /home/david/Documents/Estructuras/Proyectobonito/include/ConvexHull.hh

2.2. Referencia de la Clase HE_edge

La clase [HE_edge](#).

```
#include <HE_edge.hh>
```

Métodos públicos

- [HE_edge](#) ()
Método Constructor.
- [HE_edge](#) ([HE_vert](#) *, [HE_edge](#) *, [HE_face](#) *, [HE_edge](#) *)
Método Constructor con parametros de entrada.
- [~HE_edge](#) ()
Método Destructor.

Atributos públicos

- [HE_vert](#) * **vert**
Contiene la dirección del vértice perteneciente al Medio Borde.

- `HE_edge * pareja`

Contiene la dirección de la otra mitad del Medio Borde.

- `HE_face * face`

Contiene la dirección de la cara que contiene al Medio Borde.

- `HE_edge * next`

Contiene la dirección de un medio borde adyacente al Medio Borde.

2.2.1. Descripción detallada

La clase `HE_edge`.

Esta clase representa un Medio Borde.

2.2.2. Documentación del constructor y destructor

2.2.2.1. `HE_edge::HE_edge ()`

Método Constructor.

Este método contruye el Medio Borde.

2.2.2.2. `HE_edge::HE_edge (HE_vert *, HE_edge *, HE_face *, HE_edge *)`

Método Constructor con parametros de entrada.

Construye un Medio Borde utilizando datos anteriores.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `/home/david/Documents/Estructuras/Proyectobonito/include/HE_edge.hh`

2.3. Referencia de la Clase `HE_face`

La clase `HE_face`.

```
#include <HE_face.hh>
```

Métodos públicos

- `HE_face` (void)

Constructor sin parámetros.

- `HE_face` (`HE_edge *`)

Constructor con parametro `HE_edge`.*

- `~HE_face` (void)

Método Destructor.

Atributos públicos

- `HE_edge * edge`

Una variable pública.

2.3.1. Descripción detallada

La clase [HE_face](#).

Esta clase implementa una cara de la figura.

2.3.2. Documentación del constructor y destructor

2.3.2.1. HE_face::HE_face (void)

Constructor sin parámetros.

Este constructor no recibe ninguna parámetro. Es el constructor estandar.

2.3.2.2. HE_face::HE_face (HE_edge *)

Constructor con parametro HE_edge*.

Construye un objeto cara que apunta al primer medio borde de la figura.

2.3.3. Documentación de los datos miembro

2.3.3.1. HE_edge* HE_face::edge

Una variable pública.

Esta variable se encarga de guardar la direccion del primer medio borde perteneciente a la figura.

La documentación para esta clase fue generada a partir del siguiente fichero:

- /home/david/Documents/Estructuras/Proyectobonito/include/HE_face.hh

2.4. Referencia de la Clase HE_vert

La clase [HE_vert](#).

```
#include <HE_vert.hh>
```

Métodos públicos

- [HE_vert](#) (void)
Método Constructor sin parámetros.
- [HE_vert](#) (Point2D, [HE_edge](#) *)
Método Constructor con parámetros.
- [~HE_vert](#) (void)
Método Destructor.

Atributos públicos

- [Point2D](#) point
Una variable publica point ([Point2D](#))
- [HE_edge](#) * edge
*Una variable pública edge([HE_edge](#) *)*

2.4.1. Descripción detallada

La clase [HE_vert](#).

Esta clase implementa un vertice.

2.4.2. Documentación del constructor y destructor

2.4.2.1. `HE_vert::HE_vert (Point2D , HE_edge *)`

Método Constructor con parámetros.

Este constructor asigna los valores de las cordenadas del punto y la dirección de memoria del siguiente Medio Borde.

2.4.3. Documentación de los datos miembro

2.4.3.1. `HE_edge* HE_vert::edge`

Una variable pública `edge(HE_edge *)`

Esta variable contiene la direccion de memoria del proximo Medio Borde.

2.4.3.2. `Point2D HE_vert::point`

Una variable publica `point (Point2D)`

Esta variable contiene la coordenada en donde se encuentra el punto.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `/home/david/Documents/Estructuras/Proyectobonito/include/HE_vert.hh`

2.5. Referencia de la Clase Point2D

Clase [Point2D](#).

```
#include <Point2D.hh>
```

Métodos públicos

- [Point2D](#) (void)
Método constructor sin parámetros.
- [Point2D](#) (double, double)
Método constructor con Coordenada.
- [Point2D](#) (const [Point2D](#) &obj)
Método constructor de copia.
- [~Point2D](#) (void)
Método Destructor.
- double [getx](#) (void)
Método getx.
- double [gety](#) (void)
Método gety.
- void [setx](#) (double)

Método setx.

- void `setx` (double)

Método sety.

- void `sety` (double, double)

Método sett.

- `Point2D * setNext` (double, double)

Método setNext.

- void `setNext` (`Point2D *`)

Método getAm.

- int `getAm` (void)

Método get.

- `Point2D get` (int)

Método get.

- bool `operator!=` (`Point2D`)

Método !=.

- void `assignn` (int, `Point2D`)

Método assignn.

- bool `operator==` (`Point2D`)

Método ==.

- `Point2D operator-` (`Point2D`)

Método -.

- bool `onLeft` (`Point2D`)

Método onLeft.

- int `getLeftMost` (void)

Método getLeftMost.

2.5.1. Descripción detallada

Clase `Point2D`.

Esta clase implemente un punto en un plano.

2.5.2. Documentación del constructor y destructor

2.5.2.1. `Point2D::Point2D (void)`

Método constructor sin parámetros.

Este método crea un nuevo `Point2D`, cuya coordenada es (0,0) y los punteros apuntan a NULL.

2.5.2.2. `Point2D::Point2D (double , double)`

Método constructor con Coordenada.

Este metodo crea un nuevo `Point2D`, cuya coordenada es (x,y). el puntero next apunta a NULL.

2.5.2.3. `Point2D::Point2D (const Point2D & obj)`

Método constructor de copia.

Este metodo crea un nuevo `Point2D` a partir de un objeto `Point2D` ya existente. Es un nuevo objeto, pero sigue apuntando al mismo next.

2.5.3. Documentación de las funciones miembro

2.5.3.1. void Point2D::assignn (int , Point2D)

Método assignn.

Este método le asigna un [Point2D](#) a una entrada específica a partir del punto en cuestión.

2.5.3.2. Point2D Point2D::get (int)

Método get.

Este método retorna una entrada ([Point2D](#)) a elegir de la lista a partir de el punto en cuestion

2.5.3.3. int Point2D::getAm (void)

Método getAm.

Este método retorna la cantidad de elementos que están presentes en la lista.

2.5.3.4. int Point2D::getLeftMost (void)

Método getLeftMost.

Este método retorna la entrada de la lista en la cual se encuentra el punto de coordenada x menor.

2.5.3.5. double Point2D::getx (void)

Método getx.

Este método retorna la coordenada x del punto en cuestión.

2.5.3.6. double Point2D::gety (void)

Método gety.

Este método retorna la coordenada y del punto en cuestión.

2.5.3.7. bool Point2D::onLeft (Point2D)

Método onLeft.

Este metodo retorna TRUE si *this esta a la izquierda de el [Point2D](#) de entrada.

2.5.3.8. bool Point2D::operator!= (Point2D)

Método !=.

Este método retorna TRUE en caso de que las coordenadas de dos puntos sean diferentes.

2.5.3.9. Point2D Point2D::operator- (Point2D)

Método -.

Este método retorna un nuevo [Point2D](#) con el resultado de la resta vectorial de dos puntos.

2.5.3.10. bool Point2D::operator==(Point2D)

Método ==.

Este método retorna TRUE en caso de que las coordenadas de dos puntos sean iguales.

2.5.3.11. Point2D* Point2D::setNext (double , double)

Método setNext.

Este método crea un nuevo [Point2D](#), luego le asigna [Point2D](#) *next la dirección de el nuevo objeto. Además retorna un puntero al mismo objeto.

2.5.3.12. void Point2D::setNext (Point2D *)

Método setNext.

Este método asigna a this->next una dirección a un [Point2D](#) ya existente.

2.5.3.13. void Point2D::sett (double , double)

Método sett.

Este método asigna valores a las coordenadas x e y en el punto en cuestión.

2.5.3.14. void Point2D::setx (double)

Método setx.

Este método asigna un valor a la coordenada x del punto en cuestión.

2.5.3.15. void Point2D::sety (double)

Método sety.

Este método asigna un valor a la coordenada y del punto en cuestión.

La documentación para esta clase fue generada a partir del siguiente fichero:

- /home/david/Documents/Estructuras/Proyectobonito/include/Point2D.hh

Índice alfabético

assignn
 Point2D, 8

ConvexHull, 3

edge
 HE_face, 5
 HE_vert, 6

get
 Point2D, 8

getAm
 Point2D, 8

getLeftMost
 Point2D, 8

getx
 Point2D, 8

gety
 Point2D, 8

HE_edge, 3
 HE_edge, 4
 HE_edge, 4

HE_face, 4
 edge, 5
 HE_face, 5
 HE_face, 5

HE_vert, 5
 edge, 6
 HE_vert, 6
 HE_vert, 6
 point, 6

onLeft
 Point2D, 8

operator-
 Point2D, 8

operator==
 Point2D, 8

point
 HE_vert, 6

Point2D, 6
 assignn, 8
 get, 8
 getAm, 8
 getLeftMost, 8
 getx, 8
 gety, 8
 onLeft, 8

operator-, 8

operator==, 8

Point2D, 7

Point2D, 7

setNext, 9

sett, 9

setx, 9

sety, 9

setNext
 Point2D, 9

sett
 Point2D, 9

setx
 Point2D, 9

sety
 Point2D, 9