

Informe del primer Seminario de LP

Enrique Martínez González

1

1.1

Un template es una manera especial de escribir funciones y clases para que estas puedan ser usadas con cualquier tipo de dato y esta lo que hace es sustituir código en tiempo de compilación.

2

2.1

1. Por valor: Esto quiere decir que cuando el control pasa a la función, los valores de los parámetros en la llamada se copian a "objetos" locales de la función. Esto se hace moviendo todo el "objeto" para el stack.
2. Por puntero: Sirven para indicar la posición de un "objeto" en la memoria. Esta en memoria se maneja semejante al método por valor, se pasa todo este número de un lugar a otro en los llamados a funciones.
3. Por referencia: Las referencias sirven para definir "alias" o nombres alternativos para un mismo objeto. Esta en memoria se maneja de una manera especial en la que el compilador genera código que hace que el valor en la pila referencie directamente al "objeto" en la memoria.

2.2

En C++ uno maneja la memoria casi por completo. Solo el compilador se encarga de liberar la memoria que fue reservada en un contexto pero esta no fue creada por la función new.

3

3.1

Hay varios constructores implementados en la clase node, cada uno de ellos es una sobrecarga para distintos tipos de parametros a la hora de crearlos, también

está sobrecargado el operador `=` para que se pueda crear un nodo directamente asignando un valor genérico. El compilador sabe cuando llamar a cada uno ya que escoge el más restrictivo dado los parametros recibidos.

3.2

La inicialización de campos es establecer los valores que tendrán las distintas propiedades de la estructura en el momento de crearla.

3.3

4

4.1

5

5.1

El destructor es el que se encarga de liberar toda la memoria que utilizó el "objeto" asociado.

5.2

6

6.1

6.2

6.3

6.4

Estas son funciones que devuelven objetos que estos no se pueden modificar, un ejemplo de estas es visible en la función `Length()`, en la cual se devuelve una referencia a la propiedad `length` pero esta no se puede modificar.

6.5

Las excepciones se lanzan usando *throw* y se capturan usando los métodos de `try` y `catch()` en donde si se lanza una excepción en el `try`, esta es recogida en los parametros del `catch`.

7

7.1

7.2

7.3

7.4

El tipo de retorno debe ser por valor ya que este solo guarda los punteros a los nodos first y last, y el tamaño de la linkedlist. Los nodos fueron creados en el heap por lo que nunca se borraron al terminar el scope en este constructor.

8

8.1

9

9.1