



módulo cpp 06

dormir2 · 23 de noviembre de 2021

seguir

♥ 0

cpp

42Seúl



▼ Vista de la lista

15 / 15





	-49%		-20%
<div>Caja de almacenamiento NAS 8 HDD sin herramientas, co...</div> <div>€395.23 €197.62</div>			

ex00

float: 0.0f, -4.2f, 4.2f... y +inff, -inff, nanf
double: 0.0, -4.2, 4.2... y +inf, -inf , abuela

prueba

Examples.

```
./convert 0
char: Non displayable
int: 0
float: 0.0f
double: 0.0
./convert nan
char: impossible
int: impossible
float: nanf
double: nan
./convert 42.0f
char: '*'
int: 42
float: 42.0f
double: 42.0
```

ex01

uintptr_t serializar (ptr de datos); Deserializar
datos (uintptr_t raw);

Al implementar los dos métodos, el objetivo del proyecto es convertir literalmente la clase Data a
uintptr_t y
viceversa.

El tipo uintptr_t es un tipo entero que contiene un valor convertido de un puntero nulo y se puede
volver a convertir a ese tipo con el mismo valor que el puntero original.

La conversión de tipos es posible a través de reinterpret_cast.

Todo lo que tienes que hacer es mostrar que es una especie de tipo de puntero vacío que contiene
el valor de la dirección tal como está.

```
Base* generar(nulo);
identificación nula (Base* p);
identificación nula (Base& p);
```

Hay una clase principal Base y hay clases A, B y C que la heredan. El método generar es un método que recibe aleatoriamente una de estas tres clases como clase principal base y la devuelve.

identificar es un método que encuentra qué clase entre A, B y C es el parámetro p y lo informa. Y como typeid está prohibido, es difícil saber qué clase es A, B o C usando métodos generales.

Esto se puede solucionar utilizando el down casting. En pocas palabras, todo lo que tiene que hacer es reducir la instancia actualmente definida como clase principal base y verificar si es una clase A, B o C.

Puedes usar Dynamic_cast.



seguir

dormir2



Publicación anterior

módulo cpp 05

		-18%		-9%		-25%
		-57%	-20%		-8%	-62%
	-10%	-9%	-7%	-57%	-18%	-9%
=						

0 comentarios

[Escribir un comentario](#)

Publicaciones que podrían interesarte

Java

[Java] Conversión de tipos (Casting)

Mientras me preguntaba sobre qué tema publicar hoy, se me ocurrió una idea después de escuchar una historia de mi hermana. Cuando escuché que a mi hermana le hicieron una pregunta sobre el método de iguales en una entrevista en su empresa, pensé: '¡Ah! Pensé: "Debería publicar sobre esto hoy". Encadenar ...

enero 28, 2021 · 3 comentarios



por **Sejin Park**

 7

[c++] Aprendamos sobre herencia y polimorfismo.

Acerca de la herencia de C++



[OOP] enlace estático, enlace dinámico

Las variables y funciones de un programa deben asignar memoria para almacenar su contenido. Antes de eso, se deben determinar los tipos y valores de variables y funciones, y determinar estos tipos y valores se denomina vinculante. En este punto, aprendamos sobre el enlace estático y el enlace dinámico.

17 de junio de 2022 · 0 comentarios

👤 por **Se Dongne**

👍 Uno

[C++] Puntero de objeto

[C++] Puntero de objeto

marzo 3, 2022 · 0 comentarios

👤 por **WestCoast**

👍 Uno

[Java] Upcasting y downcasting

Crear una variable/objeto de cualquier tipo/clase de datos es posible cuando el lado derecho tiene toda la información requerida por el tipo de datos en el lado izquierdo. La conversión significa conversión de tipo, y esta conversión de tipo, si lo piensas de acuerdo con el. arriba, cambia el lado derecho para que coincida con el tipo de datos del lado izquierdo. Es para transformar el tipo de. Por ejemplo, como se muestra a continuación, int ...

marzo 19, 2022 · 0 comentarios

<Generic>

¿Conoce y utiliza genéricos?

Luego, me di cuenta de que, aunque uso Java, me falta una reflexión y un conocimiento profundos sobre Java, así que escribí este artículo.

noviembre 21, 2023 · 0 comentarios

 por Jaemin Son

♥ 19

22.03.03 Obtener información de la sesión

Para comunicarse entre procesos, se emite una ID de sesión al conectarse al servidor durante el período desde que se reconocen entre sí mediante el intercambio de mensajes hasta que se completa la comunicación => La ID de sesión se almacena en una cookie y se envía desde el servidor al cliente. Servidor Es una de las áreas de objetos integradas que puede almacenar información en ...

marzo 3, 2022 · 0 comentarios

 por gogogogo

♥ 2



[Preparación de la entrevista] Swift + RxSwift

👤 Resumen del concepto de la entrevista para desarrolladores de iOS: Swift / RxSwift

14 de abril de 2023 · 0 comentarios

👤 por Sangwoo Kim

👍 5

[Java] Herencia y polimorfismo - polimorfismo

Polimorfismo significa que un código se implementa y ejecuta con múltiples tipos de datos, lo que significa que el mismo código puede producir múltiples resultados de ejecución. El polimorfismo, una de las características más importantes de la programación orientada a objetos junto con la ocultación y la herencia de información, es la programación orientada a objetos. Flexibilidad, reciclabilidad, mantenibilidad ...

30 de agosto de 2020 · 0 comentarios

👤 por Na Young-won

👍 2

C++ #07 Polimorfismo

07. Polimorfismo Técnica del polimorfismo El polimorfismo se refiere a la capacidad de múltiples objetos diferentes para procesar la misma función de diferentes maneras. Por ejemplo, armas como espadas, cañones, pistolas, etc. pueden realizar la misma función de "ataque" de forma diferente. image.png Entonces, sin tener que implementar realmente la función attack() en el objeto arma, es abstracto...

febrero 5, 2020 · 0 comentarios

👤 por underlier12

👍 Uno

