

Informática II

El puntero `this`

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2024 –

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

- ▶ Este puntero no es parte del objeto, esto se aprecia si se utiliza el operador `sizeof` sobre el objeto

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

- ▶ Este puntero no es parte del objeto, esto se aprecia si se utiliza el operador `sizeof` sobre el objeto
- ▶ El puntero `this` se le pasa al objeto como primer argumento implícito de cada función miembro no estática (lo hace el compilador)

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

- ▶ Este puntero no es parte del objeto, esto se aprecia si se utiliza el operador `sizeof` sobre el objeto
- ▶ El puntero `this` se le pasa al objeto como primer argumento implícito de cada función miembro no estática (lo hace el compilador)
- ▶ Se utiliza de manera implícita para acceder a los miembros de un objeto

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

- ▶ Este puntero no es parte del objeto, esto se aprecia si se utiliza el operador `sizeof` sobre el objeto
- ▶ El puntero `this` se le pasa al objeto como primer argumento implícito de cada función miembro no estática (lo hace el compilador)
- ▶ Se utiliza de manera implícita para acceder a los miembros de un objeto
- ▶ También puede utilizarse de manera explícita

El puntero `this`

Todos los objetos tienen un puntero apuntando a sí mismo llamado `this`

- ▶ Este puntero no es parte del objeto, esto se aprecia si se utiliza el operador `sizeof` sobre el objeto
- ▶ El puntero `this` se le pasa al objeto como primer argumento implícito de cada función miembro no estática (lo hace el compilador)
- ▶ Se utiliza de manera implícita para acceder a los miembros de un objeto
- ▶ También puede utilizarse de manera explícita

Ver código fuente ejemplo `fig17_07.cpp`

El puntero `this`

El tipo de puntero `this` depende del tipo del objeto y de si la función miembro donde se utiliza se declara `const`

El puntero `this`

El tipo de puntero `this` depende del tipo del objeto y de si la función miembro donde se utiliza se declara `const`

1. En una función miembro no constante de la clase `Hora` el puntero `this` es del tipo `Hora * const`
(o sea, un puntero constante a un objeto `Hora`)

El puntero `this`

El tipo de puntero `this` depende del tipo del objeto y de si la función miembro donde se utiliza se declara `const`

1. En una función miembro no constante de la clase `Hora` el puntero `this` es del tipo `Hora * const`
(o sea, un puntero constante a un objeto `Hora`)
2. En una función miembro constante de la clase `Hora` el puntero `this` es del tipo `const Hora * const`
(o sea, un puntero constante a un objeto `Hora` constante)

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

```
h.establecer(20, 20, 20).imprimir();
```

(el operador punto se asocia de izquierda a derecha)

¿Qué devuelve `h.establecer(20, 20, 20)`?

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

```
h.establisher(20, 20, 20).imprimir();
```

(el operador punto se asocia de izquierda a derecha)

¿Qué devuelve `h.establisher(20, 20, 20)`?

- Otro ejemplo

```
h.establisherHora(18).establisherMinuto(30).establisherSegundo(22);
```

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

```
h.establisher(20, 20, 20).imprimir();
```

(el operador punto se asocia de izquierda a derecha)

¿Qué devuelve `h.establisher(20, 20, 20)`?

- Otro ejemplo

```
h.establisherHora(18).establisherMinuto(30).establisherSegundo(22);
```

Funciones miembros que retornan el puntero `this`

```
1  Hora &Hora::establisher(int h, int m, int s)
2  {
3      establecerHora(h);
4      establecerMinuto(m);
5      establecerSegundo(s);
6      return *this; // permite la cascada
7  }
```

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

```
h.establisher(20, 20, 20).imprimir();
```

(el operador punto se asocia de izquierda a derecha)

¿Qué devuelve `h.establisher(20, 20, 20)`?

- Otro ejemplo

```
h.establisherHora(18).establisherMinuto(30).establisherSegundo(22);
```

Funciones miembros que retornan el puntero `this`

```
1 Hora &Hora::establisher(int h, int m, int s)
2 {
3     establecerHora(h);
4     establecerMinuto(m);
5     establecerSegundo(s);
6     return *this; // permite la cascada
7 }
```

(`this` es un puntero y `*this` es un objeto; el objeto se asigna a una referencia)

El puntero `this`

- El puntero `this` se puede utilizar para poder realizar llamadas a funciones miembro en cascada. Supongamos un objeto `h` de la clase `Hora`

```
h.establisher(20, 20, 20).imprimir();
```

(el operador punto se asocia de izquierda a derecha)

¿Qué devuelve `h.establisher(20, 20, 20)`?

- Otro ejemplo

```
h.establisherHora(18).establisherMinuto(30).establisherSegundo(22);
```

Funciones miembros que retornan el puntero `this`

```
1  Hora &Hora::establisher(int h, int m, int s)
2  {
3      establecerHora(h);
4      establecerMinuto(m);
5      establecerSegundo(s);
6      return *this; // permite la cascada
7  }
```

(`this` es un puntero y `*this` es un objeto; el objeto se asigna a una referencia)

Ver código fuente ejemplo `fig17_08.cpp` (modificado)

