Guía de Estudio: Sobrecarga

# Sobrecarga de Métodos

1. ¿La sobrecarga se resuelve en tiempo de ejecución o en tiempo de compilación? Compilación.
2. ¿Los métodos estáticos pueden ser sobrecargados? Sí.
3. ¿Se puede sobrecargar un método basados en un tipo de retorno diferente?  
   No.
4. ¿Se tiene en cuenta el nombre de los parámetros para la sobrecarga?  
    No.
5. ¿Qué técnica es una alternativa a la sobrecarga de métodos en los lenguajes que no la soportan?   
   Parámetros opcionales.
6. ¿Qué debe cambiar para que la sobrecarga de un método sea válida?  
    El tipo de dato, número u orden de los parámetros.
7. Si tenemos diferentes versiones de un mismo método, ¿cómo evitamos repetir código?   
   Llamamos del método más complejo al más simple.
8. ¿Los métodos pueden tener el mismo nombre que otros elementos de una misma clase? (atributos, propiedades, etc).   
   No.
9. Repasar los ejercicios en la PPT “04-Sobrecarga.pptx”.  
   Para parcial.
10. Para después de las clases de polimorfismo: ¿Cuál es la diferencia entre sobrecargar (overload) y sobrescribir (override)?   
    Para parcial.

# Sobrecarga de Constructores

1. ¿Se pueden sobrecargar los constructores estáticos?   
   No. Porque se llama primero sólo al crear la clase.
2. ¿Se puede llamar a un constructor estático con el operador “this”?  
   No. No pueden ser llamados desde ningún ámbito.
3. ¿Se puede llamar a constructores de otras clases con el operador “this”?   
   No. Solo se pueden llamar sobrecargas de constructores de la misma clase.
4. ¿Se puede sobrecargar a un constructor privado?  
   Sí. Sólo desde adentro de la clase.
5. ¿Existe alguna alternativa al operador “this” a la hora de evitar repetir código o tareas en común entre los constructores? ¿Cuál?  
   Sí. Pueden llamar un método dentro de los contructores.
6. Indique tres ejemplos distintos de sobrecargas para el constructor de la clase Persona sin repetir el criterio utilizado (ej. Cambiar el número de parámetros sólo se puede usar una vez):  
   public Persona (string nombre, string apellido, int edad) {}

1. ¿Sólo los métodos y las clases pueden ser declarados como estáticos?

2. ¿Puedo tener miembros estáticos en clases no-estáticas?

3. ¿Puedo tener miembros no-estáticos en clases estáticas?

Sí

4. ¿Necesito instanciar un objeto de esa clase para llamar a sus métodos

estáticos?

5. ¿Cuántas copias existen de un campo estático y cuántas de uno no-estático

cuando se crearon 5 instancias de la clase?

6. ¿Se puede acceder a miembros no-estáticos desde un método estático de la misma clase?

No.

7. ¿Se puede acceder a miembros estáticos desde un método de instancia?

Sí.

8. ¿Se puede acceder a la instancia de una clase desde un método estático de la misma? (La instancia de una clase seria la mismo objeto)

9. ¿Se pueden declarar variables estáticas locales (dentro de un método) en C#?

Sí, pero es al pedo.

10. ¿Se puede declarar constantes estáticas (static const)?

No. Estático es modificable, constante nunca va a cambiar de valor.

11. De dos ejemplos de métodos estáticos que pertenezcan a las clases del

framework:

Sleep, ReadLine

12. De un ejemplo de un método de instancia que pertenezca a las clases del

framework:

toString,