

UNIVERSIDAD MADERO PUEBLA



Ingeniería de software

Proyecto Heap vs Struct

Desarrollo de software en plataformas propietarias 2

Alumno: Emigdio Espinosa Jasso.

Matricula: 20086534.

Carrera: Ingeniería de Software.

Materia: Desarrollo de software en plataformas propietarias 2

Maestro: Ing. Leticia Estrada González

|  |  |  |  |
| --- | --- | --- | --- |
| **CUADRO COMPARATIVO** | | | |
| **Variable** | **Struct** | **Class** | **Notas** |
| ¿Es un tipo de referencia? | No | Si |  |
| es un tipo de valor? | Si | No |  |
| Puede tener tipos anidados (enumeración, clase, estructura)? | Si | Si |  |
| ¿Puede tener constantes? | Si | Si |  |
| ¿Puede tener campos? | Si | Si |  |
| ¿Puede tener propiedades? | Si | Si | Campos de instancia estructura no se pueden inicializar, se inicializará automáticamente al valor por defecto. |
| Puede tener indexadores | Si | Si |  |
| Puede tener métodos | Si | Si |  |
| Puede tener eventos | Si | Si |  |
| Puede tener miembros estáticos | Si | Si | Las estructuras, al igual que las clases, pueden tener eventos, pero se debe tener cuidado de que no se suscribe a una copia de una estructura en lugar de la estructura que pretende. |
| (constructores, campos, métodos, propiedades, etc.)? | Si | Si |  |
| ¿Se pueden implementar interfaces? | No | Si | Las clases pueden heredar de otras clases (o de objeto por defecto). Las estructuras siempre heredan de System.ValueType y se sellan de manera implícita |
| ¿Puede sobrecargar el constructor? | Si | Si |  |
| Puede definir constructor por defecto? | No | Si | Sobrecarga de estructura del constructor no oculta constructor predeterminado |
| ¿Se pueden sobrecargar los operadores? | No | Si | El constructor predeterminado  inicializa struct todos los campos  de instancia a los valores por  defecto y no se puede cambiar. |
| puede ser genérico? | Si | Si |  |
| puede ser parcial? | Si | Si |  |
| Se puede sellar? | Si | Si |  |
| Puede ser referenciado en instancias? | Siempre | Si | Las estructuras siempre están  sellados y no se pueden tener  structs derivadas. |
| Miembros que usan  palabra clave? necesitan nuevo operador para crear instancia? | No | Si | Clases de C # deben instanciar  usando new. Sin embargo, las  estructuras no requieren esto.  Mientras que New se puede utilizar  en una estructura que llamar a un  constructor, puede optar por no  utilizar new e init los campos a ti  mismo, pero hay que init todos los  campos y los campos debe ser público! |
|  |  |  |  |

HEAP VS STEACK

HEAP

El montículo libre, zona libre, almacenamiento libre o heap es una estructura dinámica de datos utilizada para almacenar datos en ejecución. A diferencia de la pila de ejecución que solamente almacena las variables declaradas en los bloques previo a su ejecución, el heap permite reservar memoria dinámicamente, es decir, es el encargado de que la «magia» de la memoria dinámica ocurra. Las variables globales y estáticas también son almacenadas en él.

Como esta estructura de datos no sigue ninguna metodología, es un poco complicado mantener el control de los bloques de memoria reservados puesto que se pueden ocupar y liberar espacios en cualquier momento.

STEACK

La pila de llamadas, pila de ejecución, pila de función, pila de control, pila de tiempo de ejecución o simplemente call stack es una estructura dinámica de datos que almacena la información sobre las subrutinas activas de un programa en ejecución. Cuando hacemos una llamada a una función, un bloque en el tope de la pila es reservado para guardar las variables locales junto con algunos datos necesarios para garantizar el funcionamiento adecuado de la estructura (como la dirección a la que tendrá que retornar el hilo de ejecución cuando termine la función). Después de retornar, el bloque de la pila que ocupaba el llamado, se libera para poder utilizarse más adelante de ser necesario.

Se llama pila de ejecución porque es una estructura de datos que funciona bajo el concepto de LIFO (last in first out). Esto hace que sea más sencillo mantener el control de los bloques que deben ser liberados, puesto que será aquel que esté en el tope de la pila.

Pueden existir múltiples pilas de ejecución en un programa cuando este es multihilo; cada hilo tiene su propia pila con la que mantendrá el control de sus llamadas a funciones.

Referencias

Diferencia entre Class y Struct — documentación de Apuntes Informatica - 1.0.0. (s. f.). Recuperado 15 de octubre de 2022, de <https://apuntes-snicoper.readthedocs.io/es/latest/programacion/csharp/diferencia_class_and_struct.html>

Ruelas, U. (2017, 16 mayo). Diferencias entre heap y stack. Recuperado 15 de octubre de 2022, de <https://codingornot.com/diferencias-entre-heap-y-stack>

Stack vs.Heap: conozca la diferencia || ebooksonline.es. (2020, 1 enero). Ebooks Online. Recuperado 15 de octubre de 2022, de https://ebooksonline.es/stack-vs-heap-conozca-la-diferencia/