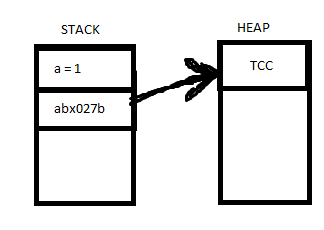
* Phân biệt Stack - Heap

-Bộ nhớ Stack được dùng để lưu trữ các biến cục bộ trong hàm, tham số truyền vào.

-Bộ nhớ Heap được dùng để lưu trữ vùng nhớ cho những biến con trỏ được cấp phát động.

Ví dụ: int a = 1;

String name = “TCC”



* Phân biệt Value Type vs Reference Type

**Value Type:**Data type (kiểu dữ liệu) là một value type (kiểu giá trị) lưu trữ data value trong momery space của riêng nó. Có nghĩa variables của data type chứa (directly) trực tiếp giá trị của variables. Vì vậy khi gán trị từ variable first cho variable other và variables other thay đổi giá trị thì không ảnh hưởng đến variable first.

**Reference Type:** khác với value type, nó không lưu giá trị theo kiểu trực tiếp (directly). Thay vào đó reference type lưu trữ address của vùng nhớ nơi dữ liệu được lưu trữ. Nói một cách khác reference type chứa một pointer trỏ đến một vị trí bộ nhớ chứa dữ liệu.

* Phân biệt var - dynamic

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **VAR** | **DYNAMIC** |
| Phải khởi tạo giá trị khi khai báo | Bắt buộc | Không bắt buộc |
| Sử dụng để làm kiểu trả về hoặc tham số cho hàm | Không | Có |
| Có khả năng ép kiểu qua lại với các kiểu dữ liệu khác | Không | Có |
| Thời điểm xác định kiểu dữ liệu thực sự | Xác định ngay tại khai báo thông qua giá trị được gán vào | Xác định ngay tại khai báo thông qua giá trị được gán vào |

* Phân biệt Boxing - Unboxing

- Boxing là kiểu chuyển đổi ngầm từ **Value Types** sang **Reference Types**, vì là kiểu tham chiếu nên nó sẽ lưu giá trị trên một Heap có kiểu tương ứng ban đầu.  
- Unboxing là di chuyển dữ liệu ngược lại từ **Reference Type** sang **Value Types**.

Ví dụ:

-Boxing: Chuyển đổi kiểu **Int** sang kiểu **tham chiếu**

int a = 1;

object o = a;

Console.WriteLine(o);

* Output: 1

- Unboxing: Chuyển đổi kiểu tham chiếu **Object** sang kiểu **Int**

object o = 1;

int a = Convert.ToInt32(o);

Console.WriteLine(a);

* Output: 1