**栈：**

栈使用物理内存，地址分配由高到低，后进先出，如分配800k的栈内存给程序，地址由800000开始向下分配

当程序定义一个局部变量时，会在堆中分配空间，如果是对象，存的是地址（堆地址），否则存的是值

**堆：**

堆使用虚拟内存，每个程序有4g的虚拟内存，但高2g的地址被系统占用，地址分配由低到高

当程序实例一个对象时，对象会被分配到堆中，如果对象的引用给了局部变量，则这个局部变量被分配到栈中，存储对象的堆地址

**垃圾回收：**

System.GC类，.net下自动回收没有引用的堆内存

**非托管资源释放：**

拥有非托管资源的对象应该继承IDisposable接口，实现Dispose()方法中释放资源

手动调用释放：

ResourceHolder resourceHolder = new ResourceHolder();

resourceHolder.Dispose();

使用using关键字：

using (ResourceHolder resourceHolder = new ResourceHolder()) {

}

析构函数中调用释放：

~ResourceHolder()

{

Dispose();

}

示例代码：

public class ResourceHolder : IDisposable

{

private bool isDisposabled = false;

~ResourceHolder()

{

// 释放自身资源

Dispose(false);

}

public void Dispose()

{

// 释放所有资源

Dispose(true);

}

/// <summary>

/// 释放资源

/// </summary>

/// <param name="disposing"></param>

protected virtual void Dispose(bool disposing)

{

if (isDisposabled == true)

return;

// 释放自身的资源

isDisposabled = true;

if (disposing == false)

return;

// 释放所引用的对象的资源（调用所有引用对象的 Dispose()）

}

}