

C# 不安全代码

当一个代码块使用 **unsafe** 修饰符标记时，C# 允许在函数中使用指针变量。**不安全代码**或非托管代码是指使用了**指针**变量的代码块。

指针变量

指针 是值为另一个变量的地址的变量，即，内存位置的直接地址。就像其他变量或常量，您必须在使用指针存储其他变量地址之前声明指针。

指针变量声明的一般形式为：

```
type* var-name;
```

下面是指针类型声明的实例：

实例	描述
int* p	p 是指向整数的指针。
double* p	p 是指向双精度数的指针。
float* p	p 是指向浮点数的指针。
int** p	p 是指向整数的指针的指针。
int*[] p	p 是指向整数的指针的一维数组。
char* p	p 是指向字符的指针。
void* p	p 是指向未知类型的指针。

在同一个声明中声明多个指针时，星号 ***** 仅与基础类型一起写入；而不是用作每个指针名称的前缀。 例如：

```
int* p1, p2, p3;    // 正确
int *p1, *p2, *p3;  // 错误
```

下面的实例说明了 C# 中使用了 **unsafe** 修饰符时指针的使用：

实例

```
using System;
namespace UnsafeCodeApplication
{
    class Program
    {
```

```
static unsafe void Main(string[] args)
{
    int var = 20;
    int* p = &var;
    Console.WriteLine("Data is: {0} ", var);
    Console.WriteLine("Address is: {0}", (int)p);
    Console.ReadKey();
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Data is: 20
Address is: 99215364
```

您也可以不用声明整个方法作为不安全代码，只需要声明方法的一部分作为不安全代码。下面的实例说明了这点。

使用指针检索数据值

您可以使用 `ToString()` 方法检索存储在指针变量所引用位置的数据。下面的实例演示了这点：

实例

```
using System;
namespace UnsafeCodeApplication
{
    class Program
    {
        public static void Main()
        {
            unsafe
            {
                int var = 20;
                int* p = &var;
                Console.WriteLine("Data is: {0} ", var);
                Console.WriteLine("Data is: {0} ", p->ToString());
                Console.WriteLine("Address is: {0} ", (int)p);
            }
            Console.ReadKey();
        }
    }
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Data is: 20
Data is: 20
Address is: 77128984
```

传递指针作为方法的参数

您可以向方法传递指针变量作为方法的参数。下面的实例说明了这点：

实例

```
using System;
namespace UnsafeCodeApplication
{
    class TestPointer
    {
        public unsafe void swap(int* p, int *q)
        {
            int temp = *p;
            *p = *q;
            *q = temp;
        }

        public unsafe static void Main()
        {
            TestPointer p = new TestPointer();
            int var1 = 10;
            int var2 = 20;
            int* x = &var1;
            int* y = &var2;

            Console.WriteLine("Before Swap: var1:{0}, var2: {1}", var1, var2);
            p.swap(x, y);

            Console.WriteLine("After Swap: var1:{0}, var2: {1}", var1, var2);
            Console.ReadKey();
        }
    }
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Before Swap: var1: 10, var2: 20
```

```
After Swap: var1: 20, var2: 10
```

使用指针访问数组元素

在 C# 中，数组名称和一个指向与数组数据具有相同数据类型的指针是不同的变量类型。例如，`int *p` 和 `int[] p` 是不同的类型。

您可以增加指针变量 `p`，因为它在内存中不是固定的，但是数组地址在内存中是固定的，所以您不能增加数组 `p`。

因此，如果您需要使用指针变量访问数组数据，可以像我们通常在 C 或 C++ 中所做的那样，使用 **fixed** 关键字来固定指针。

下面的实例演示了这点：

实例

```
using System;
namespace UnsafeCodeApplication
```

```
{
    class TestPointer
    {
        public unsafe static void Main()
        {
            int[] list = {10, 100, 200};
            fixed(int *ptr = list)

            /* 显示指针中数组地址 */
            for ( int i = 0; i < 3; i++)
            {
                Console.WriteLine("Address of list[{0}]=1",i,(int)(ptr + i));
                Console.WriteLine("Value of list[{0}]=1", i, *(ptr + i));
            }
            Console.ReadKey();
        }
    }
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Address of list[0] = 31627168
Value of list[0] = 10
Address of list[1] = 31627172
Value of list[1] = 100
Address of list[2] = 31627176
Value of list[2] = 200
```

编译不安全代码

为了编译不安全代码，您必须切换到命令行编译器指定 `/unsafe` 命令行。

例如，为了编译包含不安全代码的名为 `prog1.cs` 的程序，需在命令行中输入命令：

```
csc /unsafe prog1.cs
```

如果您使用的是 Visual Studio IDE，那么您需要在项目属性中启用不安全代码。

步骤如下：

- 通过双击资源管理器（Solution Explorer）中的属性（properties）节点，打开**项目属性（project properties）**。
- 点击 **Build** 标签页。
- 选择选项**"Allow unsafe code"**。



2 篇笔记

[写笔记](#)

fixed关键字

由于C#中声明的变量在内存中的存储受垃圾回收器管理；因此一个变量（例如一个大数组）有可能在运行过程中被移动到内存中的其他位置。如果一个变量的内存地址会变化，那么指针也就没有意义了。

解决方法就是使用fixed关键字来固定变量位置不移动。

```
static unsafe void Main(string[] args)
{
    fixed(int *ptr = int[5]) { //... }
}
```

stackalloc

在unsafe不安全环境中，我们可以通过stackalloc在堆栈上分配内存，因为在堆栈上分配的内存不受内存管理器管理，因此其相应的指针不需要固定。

```
static unsafe void Main(string[] args)
{
    int *ptr = stackalloc int[1] ;
}
```

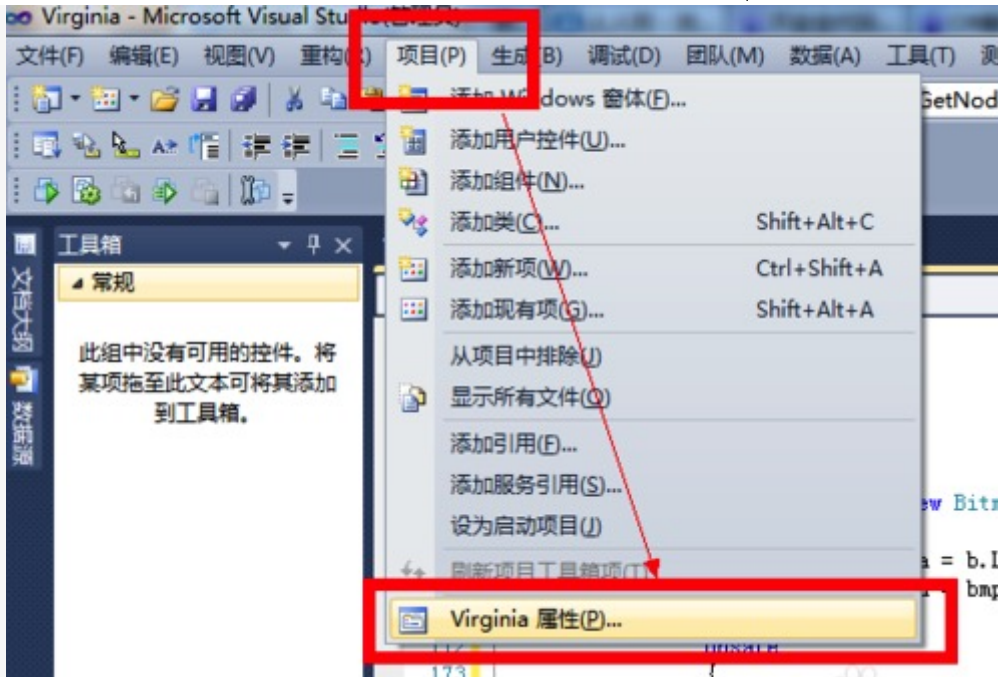
袖里藏云 2年前 (2017-07-25)



添加关键字 unsafe 后的代码会被 VS 编译环境报错。报错信息为“**不安全代码只会在使用 /unsafe 编译的情况下出现**”。

解决方法：

- 1.此时需要在上方的“**项目**”菜单中找到 **(项目标识符) 属性** 项。单击进入项目属性设置界面。



- 2. 在项目属性界面中找到 "生成" 选项卡。
- 3. 在 "生成" 选项卡中找到 "允许不安全代码"，勾选该项。



- 4. 此时再编译程序，就可以完全解决此问题了。
- 或者在 DOS 下使用编译命令如：`csc /unsafe Unsafe.cs` (Unsafe.cs 只是举例用的一个源代码文件)。

草芦卧龙 10个月前 (05-29)