

Csharp Note

ianaesthetic

September 28, 2017

1 ('□') ^ _ _

1.1 创建图形程序

，本小节在使用的是「空白应用(通用windows应用)」。在资源管理器中，.xmal文件决定了GUI，其视图有两部分：代码和框体部分。在vs2017中，右侧的「工具箱」中含有可以放到窗体上的各种控件。在创建了一个Textblock之后，右侧的「属性」窗口可以用于具体设置组件的属性。

2 变量，操作符和表达式

没有被赋值的局部变量只有在赋值后才能使用，否则被认为是编译错误。在赋值float时，数值后面一定要加上F来表示这是一个float。默认情况下是一个double值，而在csharp中不能够出现类型不同的赋值。float 和 double 在csharp中也做 mod操作。

var类型可以用于隐式变量赋值自动决定变量的类型。

3 方法和作用域

使用表达式主体方法：当方法非常简单的时候，可以简化为单个表达式构成的方法。如果有表达式有返回值则返回它，如果没有则方法自动为void方法

```
int AddValue(int leftHandSide, int rightHandSide)
=> leftHandSide + rightHandSide;
```

4 ('□') ^ _ _

5 ('□') ^ _ _

6 ('□') ^ _ _

7 类和对象

类实例的创建需要用到new + 构造函数

```
c = new Circle(r);
```

静态的理解：成员被声明为静态的时候，静态作为类的一部分而非是实例的一部分，所有的访问都依赖于类名称来访问。

静态方法：当方法声明为static, 就只能用类名字而不是实例来访问了。

```
class Math{
    public static double sqrt(double x) {
        ...
    }
}

double y = Math.sqrt(y);
```

共享字段：当字段被声明为static时，所有类型的实例同时共享同一个字段，这个字段也只能用类名字来访问。比较常见的应用是对于实例进行计数。

在调用static关键字的字段或者方法时，可以使用静态using来省略前面的类名字，使用需要谨慎。

```
using static System.Math
```

private声明的字段可以通过使用表达式定义的方法来访问

匿名类是直接new 在后面跟上一个自定义的构造函数，其中包含字段和值。拥有同样数量，名称，类型，顺序的匿名类可以相互赋值，但是可能出事。匿名类必需初始化，不可以是静态。

```
myAccountObject = new { Name = "John", Age = 47};
```

8 值和引用

8.1 引用

基元类型在声明时直接导致编译器在内存中分配内存块，而类类型在声明的时候会分配一个引用的内存块，在对于实例进行赋值时new会在内存中新开一块内存并让之前的引用能够指向这里。string也是一个类而不是一个基元类型。当下面代码出现的时候：

```
Circle c = new Circle(10);  
Circle o = c;
```

o和c实际上指向同一块地址，更改其中的任意一个都会导致另一个也被更改。

空条件操作符

```
c.?Area();
```

如果这个实例为null，则忽略这个句子。

8.2 可空类型

```
int? i = null;
```

这表示i是一个可空变量，可以被赋值，但是不能用于赋值和操作。是否为可空类型可以用HasValue成员来检测。

8.3 实参传递的引用

```
public static void Value(ref int param) {}  
public static void Mag(out int param) {  
    param = 42;  
}
```

ref表示对于参数的引用，必需使用初始化之后的值。而考虑到要通过某些方法对于值进行初始化，out关键字可以传入未被初始化参数的引用，但是在方法中必须包含对于此参数的初始化。

8.4 内存空间于System.Object

在调用方法的时候，所有的参数和局部变量都是存储在栈当中。而类型创建所需要的内存则是在堆中进行分配的。

所有类都是System.Object的派生类，同时System.Object也可以引用所有的对象和变量。用类实例对object对象的赋值相当于是创建了一个对于那个对象实例内存块的引用。

装箱是将在栈中的数据赋值到object对象时自动复制其数据到堆中的一块内存当中。

```
int i = 42;
object o = i;
```

拆箱是指对于变量用object强制转换赋值的时候将堆中相应位置的内存复制到栈中。注意只有当object和变量同类型的时候强制转换才能够生效。

```
int i = 42;
object o = i;
i = o;
```

is操作符的作用是验证对象的类型是否是自己希望的，as是is的简化版本具体的使用：

```
Circle c = new Circle(10);
...
object o1 = c;
if(o1 is Circle) {...}
...
object o2 = c;
c = o2 as Circle.
if(c != null) {...}
```

指针在Csharp中依然可以使用，但是必须将代码标记为unsafe时才能使用，同时还要指定程序「允许不安全代码」（项目属性，生成）

```
unsafe {
    swap(&x, &y);
}
...
public static unsafe void swap(int *x, int *y) {...}
```