

C#核心编程结构

元数据--描述了每一个二进制文件中定义的类型（类、结构、枚举等），以及每个类型的成员（属性、方法、事件等）。

即时编译器--将 CIL 代码编译成有意义的 CPU 指令的工具。

C#关键字都是小写的，同时命名空间、类型和成员名称（按约定）以一个大大写字母开头，中间的单词都是首字母大写。

CLS 公共语言规范

提供 sealed 来防止发生继承,For example-- sealed class MiniVan : Car

常量数据和只读数据，不同之处在于赋予只读数据的值可以在运行时决定，但是只能在构造函数的作用域内

has-a 类的关系时，那个子类吧写到父类中，如果是并行的话，会提醒他们的等级不同，出现错误。

C#的可空类型，方便数据库中数据的填写，这个暂时我还没有用到，这个倒是一个不错的特性，不用担心数据库操作的失误。

1.Main()

c#要求所有的程序逻辑都包含在一个类型定义中，Program 类中的 Main() 方法作为程序的入口点。

Program 类也称作应用程序对象，一个可执行程序可以有多个应用程序对象，但必须指定具体的入口点。

Environment: get environment info

2.Console

(1) 数据类型

object 类派生的类型： Type , String, Array, Exception, Delegate, ValueType

ValueType 类型的派生是结构或枚举，不是类

System.ValueType 重载 System.Object 定义的虚方法。来实现值类型对值操作，复制或传递等数据类型 ValueValue

(2) string: System.String

字符串拼接： +

逐字字符串： @

相等性： string 是引用类型，但是相等性操作符被定义为比较字符串对象的值

字符串是不可变的： 每一次字符串修改是分配了新的字符串

StringBuilder： System.Text

调用 stringBuilder 的成员都是直接修改对象内部的字符数据。

3.checked 关键字： 检查数据溢出的情况

unchecked: 禁止溢出抛出异常

(1) var: **隐式局部类型** 用来创建隐式类型的本地变量而不必指定具体的数据类型。编译器将自动推断实际的数据类型。

var 不是关键字，因为我们可以将之作为变量，参数的名字，但 var 标记与数据类型时，编译器视之为关键字。

可以用 var 表示任何类型，包括数组和泛型类型及自定义类型

局限：

1，只能用于方法或属性范围的本地变量，不能定义返回值，参数或自定义类型的字段。

2，必须赋初值，且不能为 null.因为 null 不能判断实际的数据类型

3，强类型，在编译时确定数据类型后将不再进行改变

4，多与 linq 查询结合使用

(2) 迭代结构

for 循环关键值只在循环域中可见

foreach 适合线性逐项遍历某个集合，不能向前或间隔等。集合可以是任何实现 IEnumerable 接口的类。

while 执行语句知道满足某个终止条件

```
while()  
{  
do  
{ }while();
```

(3) 条件结构

if/else 只能用于布尔表达式，不能用于-1/0 这样的值

布尔表达式一旦被确定为 false，其他的表达式就不会再被检查

switch

switch (number/string/enum)

```
{  
case XX: { } break;  
...  
default;  
}
```