C++中static关键字用法总结

**1．**全局静态变量

在全局变量之前加上关键字static，全局变量就被定义成为一个全局静态变量。

1）内存中的位置：静态存储区（静态存储区在整个程序运行期间都存在）

2）初始化：未经初始化的全局静态变量会被程序自动初始化为0（自动对象的值是任意的，除非他被显示初始化）

3）作用域：全局静态变量在声明他的文件之外是不可见的。准确地讲从定义之处开始到文件结尾。

好处：

定义全局静态变量的好处：

（1）不会被其他文件所访问，修改

（2）其他文件中可以使用相同名字的变量，不会发生冲突。

2．局部静态变量

在局部变量之前加上关键字static，局部变量就被定义成为一个局部静态变量。

  1）内存中的位置：静态存储区

  2）初始化：未经初始化的全局静态变量会被程序自动初始化为0（自动对象的值是任意的，除非他被显示初始化）

  3）作用域：作用域仍为局部作用域，当定义它的函数或者语句块结束的时候，作用域随之结束。

注：当static用来修饰局部变量的时候，它就改变了局部变量的存储位置，从原来的栈中存放改为静态存储区。但是局部静态变量在离开作用域之后，并没有被销毁，而是仍然驻留在内存当中，直到程序结束，只不过我们不能再对他进行访问。

当static用来修饰全局变量的时候，它就改变了全局变量的作用域（在声明他的文件之外是不可见的），但是没有改变它的存放位置，还是在静态存储区中。

3. 静态函数

在函数的返回类型前加上关键字static，函数就被定义成为静态函数。

函数的定义和声明默认情况下是extern的，但静态函数只是在声明他的文件当中可见，不能被其他文件所用。

定义静态函数的好处：

（1）其他文件中可以定义相同名字的函数，不会发生冲突

（2）静态函数不能被其他文件所用。 存储说明符auto，register，extern，static，对应两种存储期：自动存储期和静态存储期。 auto和register对应自动存储期。具有自动存储期的变量在进入声明该变量的程序块时被建立，它在该程序块活动时存在，退出该程序块时撤销。

关键字extern和static用来说明具有静态存储期的变量和函数。用static声明的局部变量具有静态存储持续期（static storage duration），或静态范围（static extent）。虽然他的值在函数调用之间保持有效，但是其名字的可视性仍限制在其局部域内。静态局部对象在程序执行到该对象的声明处时被首次初始化。

4．类内部的静态变量

在类内部的变量之前加上关键字static。

  1）内存中的位置：静态存储区

  2）初始化：要在类的外部初始化

  3）作用域：类的公有静态变量，在类的外部也可以通过 类名::变量名 调用。

注：类中不能对static数据成员进行初始化，要初始化的话必须在类外进行定义，static数据成员不是通过类构造函数进行初始化的。

5．类内部的静态函数

在类内部的函数之前加上关键字static。

1）内存中的位置：静态存储区

注：静态函数属于类而不是属于整个对象，调用静态函数没有给它对象的信息