变量

3种变量： 全局变量·静态变量·局部变量(自动变量)

# 全局变量

1. 函数外面声明
2. 可以跨文件访问
3. 可以在声明时赋上初始值
4. 如果没有赋初始值，系统自动赋值为0，或控制字符串
5. 存储位置：既非堆，也非栈，而是专门的【全局（静态）存储区static】！
6. 编译时直接声明存储在全局区

关于extern关键字

## 方式一

在A类中定义全局变量（函数外边声明），要注意的是，全局变量可以在许多地方***声明*** 为extern，但***定义***（赋初值）只能一次，定义的时候并不需要extern专门来修饰。倒是在其他不需要定义该全局变量的地方，需要extern来修饰该全局变量，声明要调用外部变量了。我们在B类想调用A类中声明和定义的全局lastNum时，并且B类也没有导入A类，这时，需要在B类中做extern float lastNum;这样声明。这样就告诉系统，我要访问其他文件中定义的这个全局变量了。***注意点***：此时，B类通过extern调用A类声明的全局变量时，***B类中并不需要导入A类，例如类似import A.h等等***。

如下：

A.m 中定义和声明全局变量

文本

中度可信度描述已自动生成

B.m 中extern并调用全局变量

文本

描述已自动生成

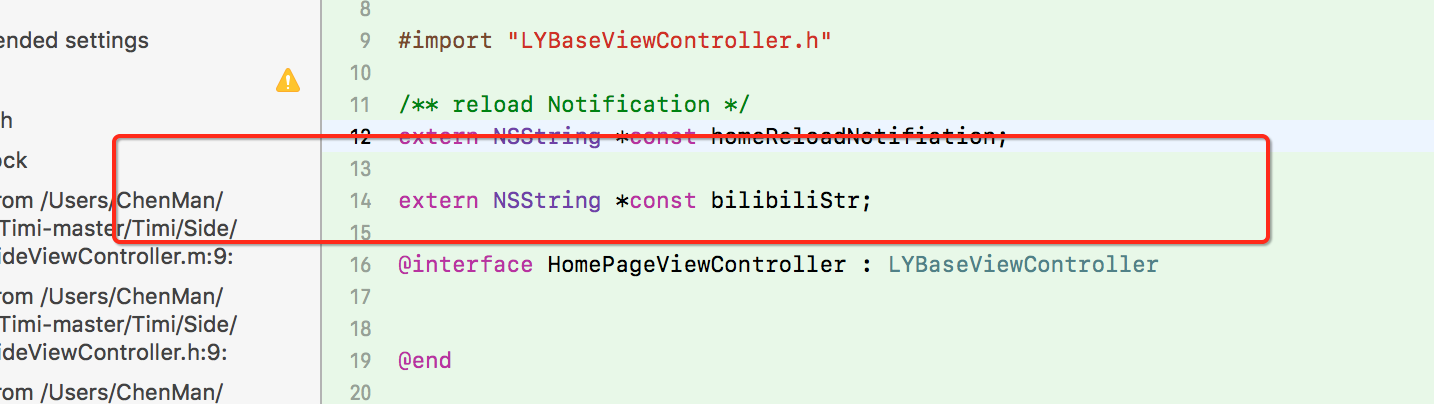
## 方式二

* 我们在A.m文件中声明和定义的全局lastNum时，同时在A.h文件中做extern float lastNum;这样的声明。在需要在B类中导入A类头文件，就不需要在B类中专门做extern float lastNum;这样的声明了。这样就告诉系统，其它访问A文件中定义的这个全局变量了，导入A类即可使用。

A.m 中声明全局变量



A.h中extern全局变量



B.m中importA.m类并调用全局变量

文本

描述已自动生成

# 静态变量

1. 函数外面 或 内部声明（即可修饰原全局变量亦可修饰原局部变量）
2. 仅声明该变量的文件可以访问
3. 可以在声明时赋上初始值
4. 如果没有赋初始值，系统自动赋值为0
5. 存储位置：既非堆，也非栈，而是专门的【全局（静态）存储区static】！

声明如下：

static float lastNum;

static float lastNum = 10.0;

static

修饰局部变量：

1. 延长局部变量的生命周期,程序结束才会销毁。
2. 局部变量只会生成一份内存,只会初始化一次。
3. 改变局部变量的作用域。

修饰全局变量

1. 只能在本文件中访问,修改全局变量的作用域,生命周期不会改
2. 避免重复定义全局变量

**优点**

1. 节省内存。静态变量只存储一处，但供所有对象使用。
2. 它的值是可以更新的。
3. 可提高时间效率。只要某个对象对静态变量更新一次，所有的对象都能访问更新后的值

方法体和类中都可以定义相同的变量名，但是形同变量名属于不同的变量对象。

静态方法只能调用静态方法和静态变量

# 局部变量（自动变量）

1. 函数内部声明
2. 仅当函数执行时存在
3. 仅在本文件本函数内可访问
4. 存储位置：自动保存在函数的每次执行的【栈帧】中，并随着函数结束后自动释放，另外，函数每次执行则保存在【栈】中

# 全局变量和静态变量的区别

与全局变量相比，静态变量存储位置一样，声明位置如果也一样（函数外部），静态变量这时跟全局变量有什么区别？就是为了限制访问范围，静态变量仅当前声明该变量文件里面的代码可以访问。而全局变量可以同一工程跨文件访问，可能会引起严重的混淆问题。