2018-08-28

2.2.变量  
 (1)初始化不是复制， 初始化的含义是创建变量时赋予其一个初始值，而赋值的含义是把对象的当前值擦出，而以一个新值来代替可。

1. 初始化的方式，例如定义一个名为units\_sold的int变量并初始化为0：
   * 1. Int units\_sold = 0
     2. Int units\_sold = {0}
     3. Int units\_sold{0}
     4. Int units\_sold(0)

花括号初始化变量的方式称为列表初始化。列表初始化一个重要的特点就是：如果我们使用列表初始化且初始化值存在丢失信息的风险，则编译器将报错。

1. 变量声明和定义的区别：
   * 1. ”声明“使得名字为程序所知，一个文件如果想要使用别处定义的名字则必须包含对那个名字的声明，不开辟内存空间。如果向声明一个变量而非定义它，则在变量名前面添加关键字”extern”，但是如果包含显示的初始化声明则失效，声明变为定义。
     2. “定义”负责创建与名字关联的实体，开辟内存空间。
     3. 变量只能被定义一次，但是可以被声明很多次。
     4. 当你第一次使用变量的时候再定义它。
2. 复合类型，指的是基于其他类型定义的类型（指针和引用）：

2018-08-30

2.3.指针和引用

2.3.1.指针：

1. 指向常量的指针(常指针)：const int \*p
2. 指针本身是常量（常量指针）：int \* const p

2.3.2.引用：

1. 引用本身不是对象，声明时必须初始化应用某个实例对象。类中的引用成员变量必须用列表初始化
2. 引用本身不是对象所以没有指向引用的指针，但是有引用指针的引用声明：int \*&p，从右向左读p是引用，引用int指针。

2.4.const限定符

(1)const 与宏不一样在于，const是运行时确定值形如：const int i = get\_size()(运行时确定)合法，而宏是预编译器阶段确定其值。另一方面宏没有类型检查，宏是简单的替换。而const则有类型检查，const变量在运行时会将所有变量替换为实际的值。

（2）const特性只会在执行改变const变量的操作时发挥作用。将const常量赋值给非const变量是合法的。

（3）默认情况下const变量只在当前文件有效，在多个文件中定义同名const变量时，这些变量是互相独立的。特别的，当在const前添加extern修饰符时，该const变量是全局的，可在多个文件中共享，且变量初始值不能为常量表达式。

（4）const引用：对常量的引用，形如：const int &r = var，引用的变量不嫩被修改。

（5）引用类型必须与其所引用对象的类型一致。因为如果引用的类型不一致而发生类型转换（其实是生成一个额外的临时变量），引用一个临时变量不是我们想要的，所以编译器将这种行为定为非法行为。

（6）可以常量引用一个非常量的值。

2018.9.3

2.4.3顶层const和底层const（p58）

1. 底层const只与指针和引用等符合类型相关的，通常指的是指针（引用）的对象是常量。
2. 顶层const不影响拷贝操作，而底层则影响。

3.1命名空间的using声明

1. 头文件不应该包含using声明，因为头文件会被拷贝到所有引用它的文件中去，重复包含会引起名字冲突。

3.2.2string对象上的操作

（1）cin从输入流中读取字符串时会忽略空白（空格符、换行符、制表符等）

（2）当希望在最终读入的字符串中保留输入时的空格符，则使用getlin()函数代替>>，gitline()函数参数是一个输入流和一个string对象,函数从给定的输入流中读入内容，直到遇到换行符（注意换行符被读入，但是会被丢弃），函数返回一个cin对象。

（3）string中的size()函数返回一个无符号类型string::size\_type，千万不要将无符号类型和int混用！！！

（4）标准模板库允许把字符字面值和字符串字面值转换成string对象，所以在需要string对象的地方也可以使用字符字面值或者字符串字面值替代。