1. explicit关键字：只能写在构造函数前面，只是针对Maker maker=10;防止该形式的代码，叫编译器不要优化成Maker maker=Maker(10)
2. new和delete
   1. new :从堆区申请空间，做初始化工作（调用类的构造函数），返回的是指针
   2. delete:在对象销毁前，会做清理工作（调用类的析构函数），然后释放空间
   3. 注意点：
      1. new的返回值不要void\*来接（因为编译器不认识void\*类型）
      2. new时有[],那么delete时也要加上[]
3. 静态成员
   1. 静态成员变量
      1. 不属于对象，属于类，所有对象共享
      2. 静态成员变量，必须在类内声明，类外初始化
      3. 用类名可以访问，用对象也可以访问
      4. 静态成员变量的生命周期为整个程序，作用域在类内
      5. 静态成员变量有权限
      6. const修饰的静态成员变量最好在类内初始化
   2. 静态成员函数
      1. 静态成员函数用来管理封装静态成员变量
      2. 静态成员函数只能访问静态成员变量
      3. 静态成员函数也有权限
      4. 普通函数可以访问静态成员变量
4. this指针
   1. 每一对象都有this指针
   2. this指针指向本对象
   3. 形参和成员变量同名时可以用this指针来区别
   4. 返回对象本身
   5. 成员变量和成员函数是分开存储
   6. 对象的大小不包括成员函数
   7. 空类的大小是一个字节
   8. const修饰的成员函数叫常函数，常函数内不能修改成员变量，但是mutable修饰的成员变量除外
5. 常对象
   1. const修饰的对象叫常对象
   2. 常对象不能改变普通的成员变量，但是mutable修饰的成员变量除外
   3. 常对象不能调用普通的成员函数，可以调用常函数
6. 友元
   1. 友元的作用：在类外访问类的私有成员
   2. 友元分：友元函数，友元类，成员函数的友元函数
   3. 在普通函数或类的前面加上friend，在类中声明
   4. 友元类不能被继续，友元是单向，友元没有传递性
7. 单列模式
   1. 什么是单例模式：一个类只能实例化一个对象
   2. 怎么实现：
      1. 把默认构造，默认拷贝构造私有化
      2. 类内声明一个静态本类的类指针
      3. 在类外定义该指针，给这个指针一对象（new）
      4. 保护该指针，要该指针私有化，然后提供获取该指针的方法（静态成员函数）