Stack(栈)和Heap(堆)

heap：是由malloc之类函数分配的空间所在地。地址是由低向高增长的。

stack：是自动分配变量，以及函数调用的时候所使用的一些空间。地址是由高向低减少的。

- 多线程和多进程中的stack和heap

  每个线程都有其自己独立的栈，它们都共享一个堆。栈是面向线程的而堆是面向进程的

  - 函数内部声明的变量在离开作用域之后就会被自动释放

static：被static修饰的变量叫做静态全局变量

```
（1）在修饰变量的时候，static 修饰的静态局部变量只执行初始化一次，而且延长了局部变量的生命周期，
（2）static 修饰全局变量的时候，这个全局变量只能在本文件中访问，不能在其它文件中访问，即便是 ext
（3）static 修饰一个函数，则这个函数的只能在本文件中调用，不能被其他文件调用。static 修饰的变量
（4）不想被释放的时候，可以使用static修饰。比如修饰函数中存放在栈空间的数组。如果不想让这个数组在
（5）考虑到数据安全性（当程序想要使用全局变量的时候应该先考虑使用 static）。
```

静态全局变量的特点

```
（1）静态变量都在全局数据区分配内存，包括后面将要提到的静态局部变量;
（2）未经初始化的静态全局变量会被程序自动初始化为0（在函数体内声明的自动变量的值是随机的，除非它
（3）静态全局变量在声明它的整个文件都是可见的，而在文件之外是不可见的。
```

被static修饰的变量在这个类中只有一份，所有对象共用这一个变量

全局变量和静态全局变量的区别

```
1）全局变量是不显式用 static 修饰的全局变量，全局变量默认是有外部链接性的，作用域是整个工程，在
2）全局静态变量是显式用 static 修饰的全局变量，作用域是声明此变量所在的文件，其他的文件即使用 ex
```

在C++中static的使用

- 变量可以被static修饰
- 函数被static修饰：与一般函数不同的是没有this指针，不能去提取和访问对象的成员。所以它只能访问静态变量（被static修饰了的类的成员）

调用static函数的方法：

1. 通过对象去调用
2. 通过类的name去调用

- 类的成员被static修饰：就会跟对象脱离，拥有了一个单独的区域。必须在类的外部进行初始化

Demo

```
#include <iostream>

```

```
class Account
{
public:
        static double m_rate;
        static void set_rate(const double &x){  m_rate = x;}

};
double Account :: m_rate;

int main(int argc,char **argv)
{
        Account :: set_rate(5.0);
        Account a;
        a.set_rate(6.0);
        return 0;

}
```

```
1、被 static 修饰的变量属于类变量，可以通过类名.变量名直接引用，而不需要 new 出一个类来
2、被 static 修饰的方法属于类方法，可以通过类名.方法名直接引用，而不需要 new 出一个类来
```

被 static 修饰的变量、被 static 修饰的方法统一属于类的静态资源，是类实例之间共享的，换言之，一处变、处处变。在 C++ 中，静态成员是属于整个类的而不是某个对象，静态成员变量只存储一份供所有对象共用。所以在所有对象中都可以共享它。使用静态成员变量实现多个对象之间的数据共享不会破坏隐藏的原则，保证了安全性还可以节省内存。静态成员的定义或声明要加个关键 static。静态成员可以通过双冒号来使用即 **<类名>::<静态成员名>**。

new关键字

在new class时

- 在C++中 new先分配内存空间，再执行构造函数

delete关键字

在delete class空间时

- 先调用析构函数再释放空间

进一步补充

把构造函数放到private中，声明一个static对象，就说明，这个对象是所有的对象都可以访问到，而且只有一个

类模板