

new delete 和 malloc free 的区别

- 1- 一个是C++里，存在析构的； 一个是C语言的
- 2- malloc 无法满足动态对象的要求
- 3- malloc和free还不能为动态对象初始化，要想初始化必须给类再写一个初始化void Initialize();的函数

之前面试的时候面试官问了我一个问题，是C语言中有了malloc和free为什么还要再有new 和 delete。说一下这两者的区别。其实之前也有看过这个东西，包括

- malloc free是C语言的库函数，而new和delete是C++的操作符。
- malloc开辟出的是一片空间，返回的是一个void *的指针，用的时候必须强制转换。而new直接返回的就是该类型的指针。但是这些都不是最为重点的。
- 最为关键的就是对于非内部数据类型的对象也就是自定义的类型而言，这两者的区别是最大的。**光用malloc free无法满足动态对象的要求。**
- 对象在创建的同时要执行构造函数，对象在消亡之前要自动执行析构函数。由于malloc free是库函数不是操作符，不在编译器控制权限之内，无法把构造函数和析构函数任务强加给malloc和free。
- 因此 C++语言有了new 和 delete来完成动态内存分配和初始化的工作。
- 对了，**malloc和free还不能为动态对象初始化，要想初始化必须给类再写一个初始化void Initialize();的函数。**
- 最后，为什么C++不淘汰掉malloc和free呢，这个很简单，因为它还要调用C的函数啥的，兼容一下下。delete可以释放malloc的内存，但不建议这么做，free不能释放new的内存，因为它无法执行析构函数，编译会出错。