某些类需要在运行时分配可变大小的内存空间，可以通过标准库容器来实现，但是可以自己进行内存管理，自己定义拷贝控制成员来管理

例子：

只管理string

实现：用allocator类来管理。

重要：

重新分配内存的过程中移动而不是拷贝元素

拷贝这些string是多余的

\*移动构造函数和std::move：

在内存不够，重新分配内存，需把旧内存的对象弄到新内存中，使用移动，而不是拷贝可以节省花销。

两种机制 使用移动而不是拷贝:

1.string等标准库类，都定义了移动构造函数，移动后仍保持有效的 可析构的状态

2.move标准库函数 utility头文件中

用move来表示希望使用string的移动构造函数，如果没用move就会用string的拷贝构造函数。

通常不为move提供一个using声明，而是直接调用std：：move，因为怕自定义的move函数优先匹配标准库的move函数。