# 10.8 总结

面向对象编程强调的是程序如何表示数据。使用OOP办法解决编程问题的第一步是根据它与程序之间的接口来描述数据，从而指定如何使用数据。然后，设计一个类来实现该接口。一般来说，私有数据成员存储信息，公有成员函数（又称为方法）提供访问数据的唯一途径。类将数据和方法组合成一个单元，其私有性实现数据隐藏。

通常，将类声明分成两部分组成，这两部分通常保存在不同的文件中。类声明（包括由函数原型表示的方法）应放到头文件中。定义成员函数的源代码放在方法文件中。这样便将接口描述与实现细节分开了。从理论上说，只需知道公有接口就可以使用类。当然，可以查看实现方法（除非只提供了编译形式），但程序不应依赖于其实现细节，如知道某个值被存储为int。只要程序和类只通过定义接口的方法进行通信，程序员就可以随意地对任何部分做独立的改进，而不必担心这样做会导致意外的不良影响。

类是用户定义的类型，对象是类的实例。这意味着对象是这种类型的变量，例如由new按类描述分配的内存。C++试图让用用户定义的类型尽可能与标准类型类似，因此可以声明对象、指向对象的指针和对象数组。可以按值传递对象、将对象作为函数返回值、将一个对象赋给同类型的另一个对象。如果提供了构造函数，则在创建对象时，可以初始化对象。如果提供了析构函数方法，则在对象消亡后，程序将执行该函数。

每个对象都存储自己的数据，而共享类方法。如果mr\_object是对象名，try\_me()是成员函数，则可以使用成员运算符句点调用成员函数：mr\_object.try\_me()。在OOP中，这种函数调用被称为将try\_me消息发送给mr\_object对象。在try\_me()方法中引用类数据成员时，将使用mr\_object对象相应的数据成员。同样，函数调用i\_object.try\_me()将访问i\_object对象的数据成员。

如果希望成员函数对多个对象进行操作，可以将额外的对象作为参数传递给它。如果方法需要显式地引用调用它的对象，则可以使用this指针。由于this指针被设置为调用对象的地址，因此\*this是该对象的别名。

类很适合用于描述ADT（抽象数据类型）。公有成员函数接口提供了ADT描述的服务，类的私有部分和类方法的代码提供了实现，这些实现对类的客户隐藏。