在 OOP 的程序设计方法中，抽象数据类型、类和对象这些概念是程序的基石。数据和其结构不再分离，它们被封装在一起，与现实世界本来面貌更接近。对象的封装性将大量的实现细节封装于内部。其好处是使程序结构的复杂性降低了，使程序更易于修改，也更容易实现代码的重用。

类有继承性和多态性两个最主要的特征。继承性是使类反映现实世界的层次结构，而多态性则反映了现实世界的复杂多样。OOP 允许开发人员通过继承来重用代码和数据。通过从以前的对象中继承，编程者可以更迅速地构造复杂的应用程序。其原因在于，编写新的代码总是会有可能带入错误，而重用已经过测试的代码可以使产生潜在错误的可能性达到最小。类的出现使程序的描述能力得到提高，使代码的真正重用成为可能。

例如，软件开发过程中，用于界面设计的工作量常常达到 70% 以上。所以就有了很多用于开发界面的函数库。但问题在于用户很难对其进行修改扩充。其原因在于，修改函数库首先必须读懂其源代码。对用户而言，这通常是困难的，并且会引进错误。但如果提供的仅仅是类库，那么，只需要简单地继承一个类，然后，重新定义或增加新的函数就行了。

综上所述，本项目拟采用OOP方法进行设计，以便程序后续的修改，以及实现代码的重用。