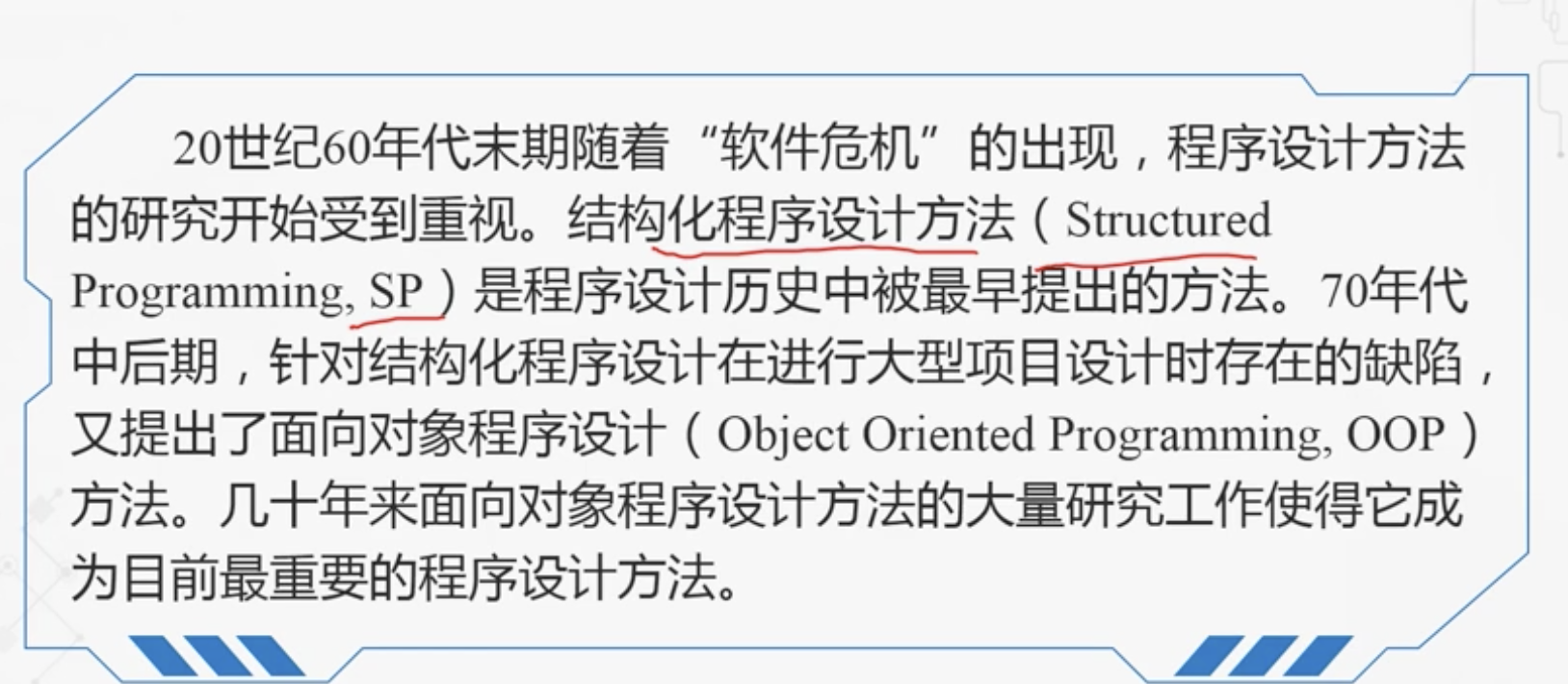
# 背景

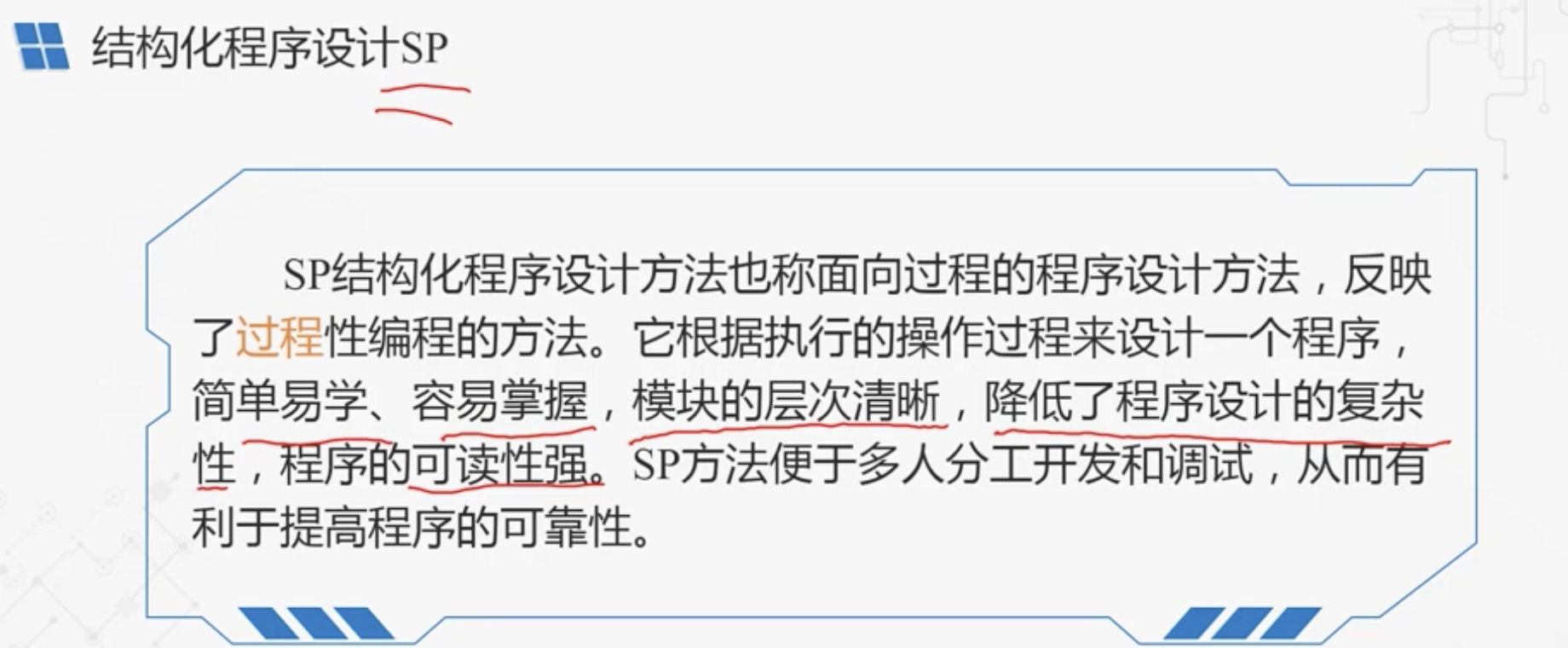


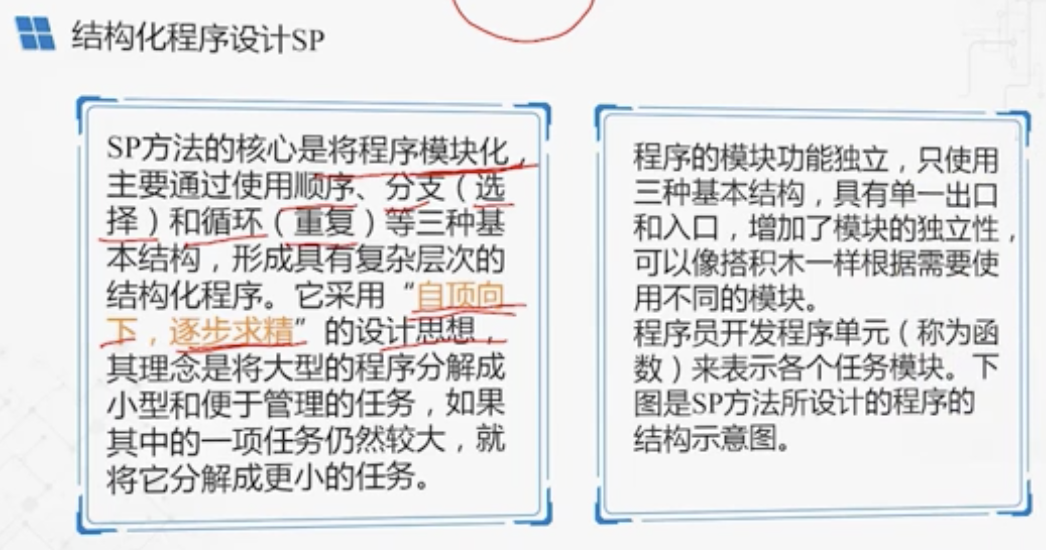
# 软件 = 程序 + 文档

# 三种不同类型的翻译程序

* 汇编程序：将汇编语言源程序翻译成目标程序；
* 编译程序：将高级语言源程序翻译成目标程序；
* 解释程序：将高级语言源程序翻译成机器指令它是边翻译边执行；

# 结构化程序设计SP





# 面向过程的程序设计方式

机器语言、汇编语言、高级语言都支持；

最初的目的：用于数学计算；

**主要工作：设计求解问题的过程**

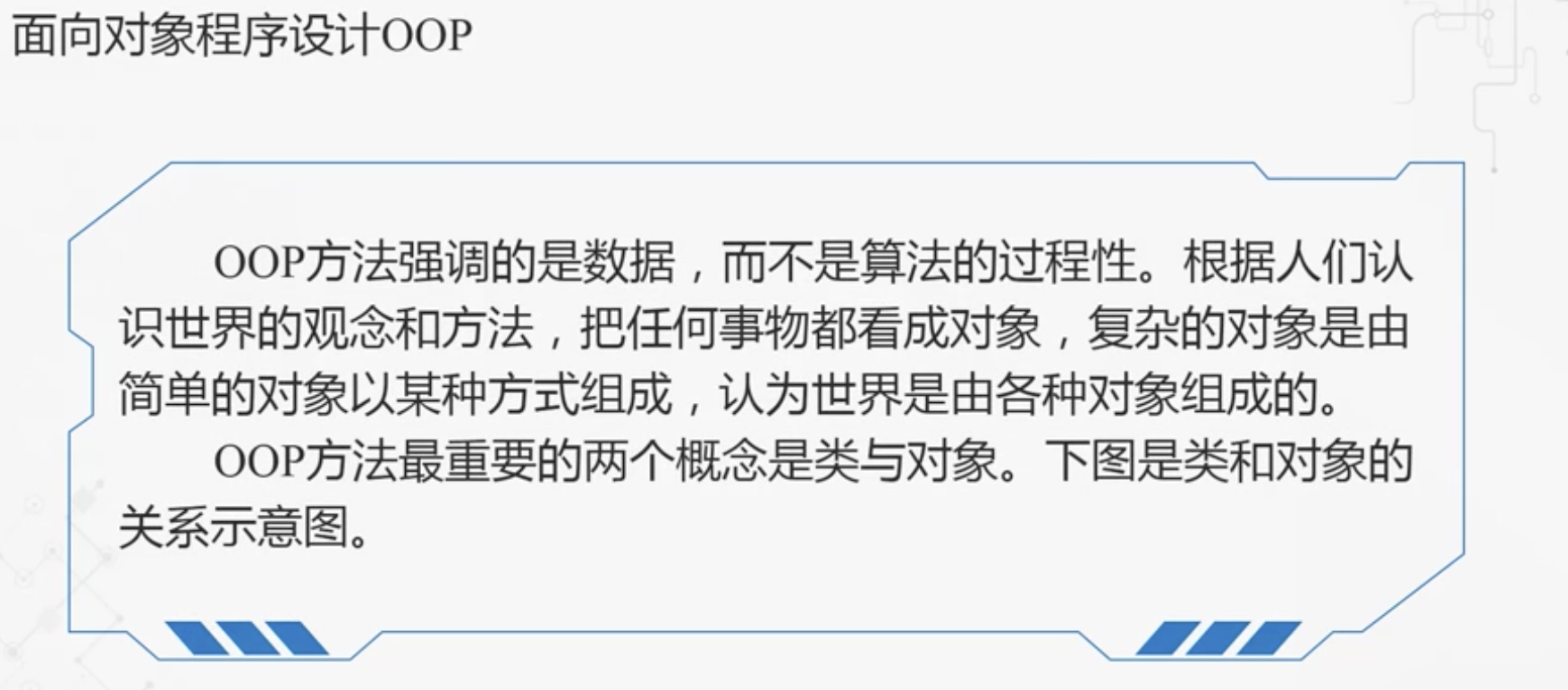
## 缺点

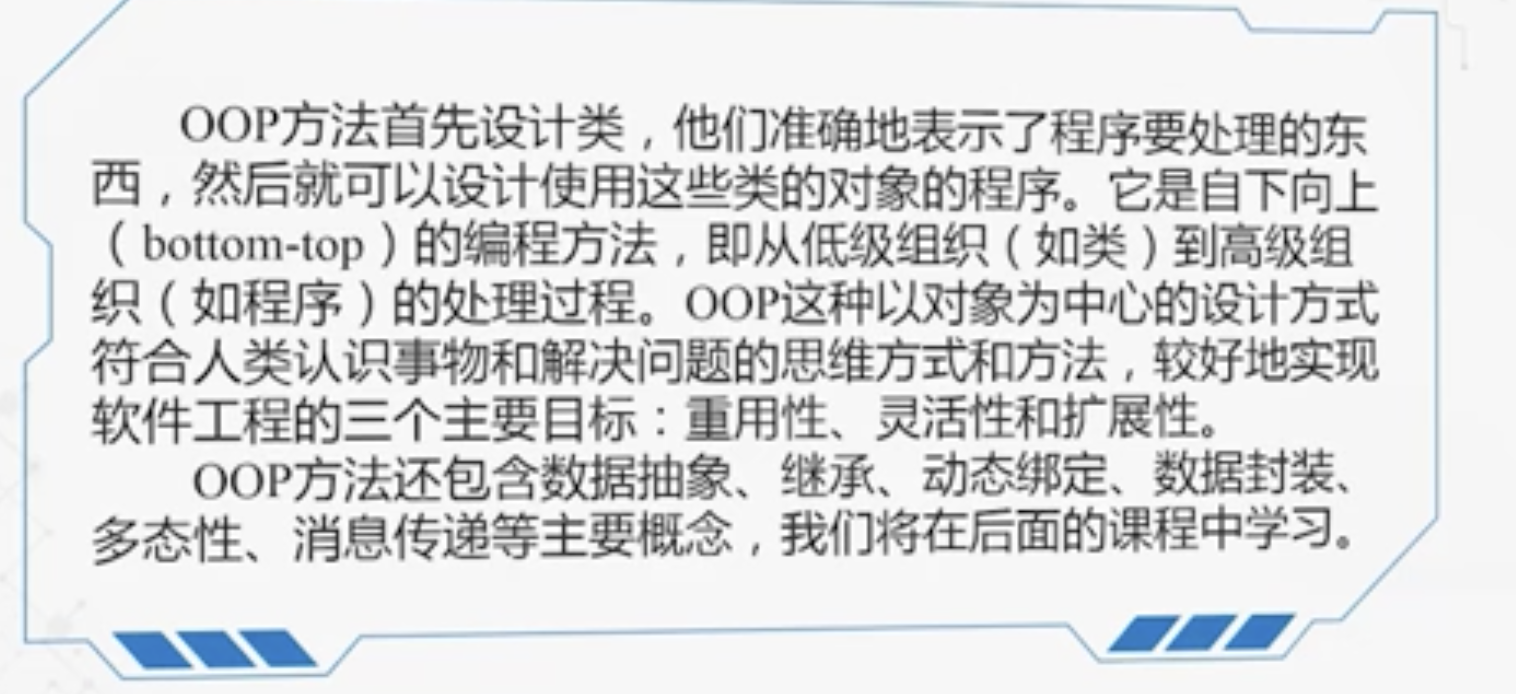
大型复杂的软件，难以用面向过程的方式编写；

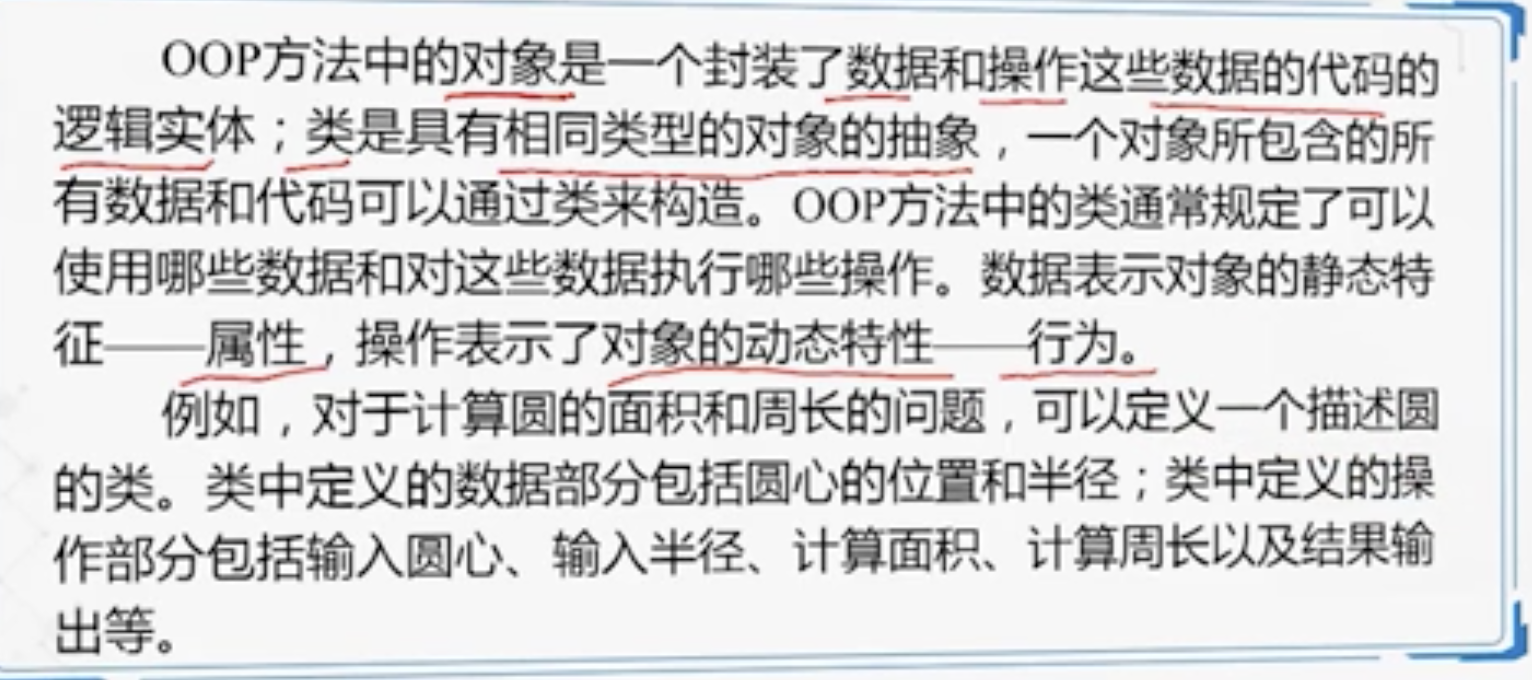
# 面向对象程序设计OOP

## 关键字理解

* 类：是分类的意思，在面向对象中，将事物进行分门别类；
* 封装：方便使用、安全性好；
* 继承：子承父业，在原来的基础上发展壮大，比自己从头开始容易得多；有复用性；
* 多态：同样的消息作用在不同对象上有可能引起不同的行为；







# 程序设计方法

* 顺序结构
* 选择结构（单选择、双选择、多选择）
* 循环结构（**当型循环（while)是先检验条件再运行，直到型循环（do…while)是先运行一次循环体）**

# 计算机语言的级别

* 二进制代码构成的机器语言
* 使用助词符的汇编语言*（便于人识别和记忆），但是这此都是抽象层次比较低的，如果写大型软件会很累*
* 使用类似英语单词和语句的高级语言

# 汇编语言

汇编语言和汇编语言的抽象程度是一样的；

* 将机器指令映射为一些助记符。如ADD、SUB、mov等；
* 抽象层次低，需要考虑机器细节

# 高级语言

* 关键字、语句容易理解；
* 有含义的数据命名和算式；
* 抽象层次较高；例如：算式：a+b+c/d;
* 屏蔽了机器的细节；例如：这样显示计算结果：cout<<a+b+c/d

***设计高级语言的目的是让语言更加符合人类的思考方式来编写程序***

# C++程序的开过过程

1. 算法设计
2. 源程序编辑（编码、写程序）
3. 编译（翻译成计算机能识别的）
4. 连接（一个大任务需要很多人来完成，需要调度他们）
5. 运行调试

# 参考资料

* *https://www.bilibili.com/video/BV1CC4y1s7np*