继承关系

当一个对象继承了一个父对象,在一个带参方法里,参数为父对象,传入一个当前的对象依然可以正常执行程序.

这种把衍生类型当作基本类型处理的过程叫做上溯造型.

对象创建和存在时间

Java使用在内存池(也叫“堆”或者“内存堆”)中动态创建对象.这种方式只有在进入运行期才会去决定对象的数量,存活时间,类型

由于存储空间的管理是运行期间动态进行的，所以在内存堆里分配存储空间的时间比在堆栈里创建的时间长得多（在堆栈里创建存储空间一般只需要一个简单的指令，将堆栈指针向下或向下移动即可）。

在内存中创建需要考虑对象的“生存时间”.程序员可用程序化的方式决定何时破坏对象,或者利用运行环境提供的一种“垃圾收集器”特性,自动寻找那些不再使用的对象,并将其清除.

单根结构

所有类最终都是从单独一个基础类继承的.

单根结构中的所有对象都有一个通用接口，所以它们最终都属于相同的类型。

违例控制：解决错误

Java违例控制模块是从一开始就封装好的(也就是异常的抛出).

如果没有自己写一些代码来正确地控制违例，就会得到一条编译期出错提示。这样可保证程序的连贯性，使错误控制变得更加容易.

多线程

一个基本的概念就是同时对多个任务加以控制。

多线程处理一个常见的例子就是用户界面。利用线程，用户可按下一个按钮，然后程序会立即做出响应，而不是让用户等待程序完成了当前任务以后才开始响应。

如果有多个线程同时运行，而且它们试图访问相同的资源，就会遇到问题。

所以使用期间必须进入锁定状态。一个线程可将资源锁定，在完成了它的任务后，再解开（释放）这个锁，使其他线程可以接着使用同样的资源。

分析与设计

1. 拟出一个计划

第一步是决定在后面的过程中采取哪些步骤。

1. 要制作什么？

这个阶段称为“建立需求分析和系统规格”。

感觉一下大概需要花多长的时间，然后将这个时间加倍，再加上 10％。

1. 如何构建？

在这一阶段，必须拿出一套设计方案，并解释其中包含的各类对象在外观上是什么样子，以及相互间是如何沟通的。

此时可考虑采用一种特殊的图表工具, 使用OOP的设计方式。

1. 开始创建

无论它有多么简要，而且在正式编码前掌握了正确的设计结构，所以会发现接下去

的工作比一开始就埋头写程序要简单得多。

全面的思考、周密的准备、良好的构造不仅使程序更易构建与调试，也使其更易理

解和维护，而那正是一套软件赢利的必要条件。

1. 校订

“维护”表示从“保持它按设想的轨道运行”、“加入客户从前忘了声明的功能”或者更传统的“除掉暴露出来的一切臭虫”等等意思。所以换言之，“你第一次做的东西并不完善，所以需为自己留下一个深入学习、认知的空间，再回过头去作一些改变”。

将整个项目达到理想的状态。

我们需要不断地对比自己的需求，了解系统是否自己实际所需要的。

它也意味着代码的内部结构应当尽善尽美。至少，我们应能感觉出整个结构都能良

好地协调运作。没有笨拙的语法，没有臃肿的对象，也没有一些华而不实的东西。

除此以外，必须保证程序结构有很强的生命力。