1. 什么是软件危机？它有哪些典型表现？为什么会出现软件危机？结合日常生活，简要描述你所知道的软件危机有哪些？

<1>.软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。

<2>.软件危机的典型表现：

（1）对软件开发成本和进度的估计常常很不准确；

（2）软件成本在计算机系统总成本中所占的比例逐年上升；

（3）软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势 。

（4）软件通常没有适当的文档资料；

（5）软件常常是不可维护的；

（6）软件产品的质量往往靠不住；

（7）用户对“已完成的”软件系统不满意的现象经常发生；

<3>.产生软件危机的原因：

（1）用户对软件需求的描述不精确。

（2）软件开发人员对用户需求的理解有偏差。

（3）缺乏处理大型软件项目的经验。

（4）开发大型软件易产生疏漏和错误。

（5）缺乏有力的方法学的指导和有效的开发工具的支持。

（6）面对日益增长的软件需求，人们显得力不从心

<4>.软件危机案例：

(1)1995年，Standish Group研究机构以美国境内8000个软件项目作为调查样本，调查结果显示，有84%软件计划无法于既定时间、经费中完成，超过30%的项目于运行中被取消，项目预算平均超出189%。

**IBMOS/360操作系统案例**

(2)IBMOS/360操作系统被认为是一个典型的案例。到现在为止，它仍然被使用在360系列主机中。这个经历了数十年，极度复杂的软件项目甚至产生了一套不包括在原始设计方案之中的工作系统。OS/360是第一个超大型的软件项目，它使用了1000人左右的程序员。佛瑞德·布鲁克斯在随后他的大作《人月神话》中曾经承认，在他管理这个项目的时候，他犯了一个价值数百万美元的错误。

**美国银行信托软件系统开发案**

(3)美国银行1982年进入信托商业领域，并规划发展信托软件系统。项目原订预算2千万美元，开发时程9个月，预计于1984年12月31日以前完成，后来至1987年3月都未能完成该系统，期间已投入6千万美元。美国银行最终因为此系统不稳定而不得不放弃，并将340亿美元的信托账户转移出去，并失去了6亿美元的信托生意商机。

1. 什么是软件工程？它有哪些本质特性？怎样用软件工程消除软件危机？

(1)软件工程的定义

概括的说，软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念，原理，技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它。

(2)软件工程的本质特征

                a:软件工程关注大型程序的构造

                b:软件工程的中心课题是控制复杂性

                c：软件经常发生变化

                d:开发软件的效率非常重要

                e:和谐地合作是开发软件的关键

                f:软件必须有效地支持他的用户

                g:在软件工程领域中通常由具有一种文化背景的人替具有另一种文化背景的人创造的产品

(3)消除软件危机的方法

                a:应该对计算机软件有一个正确的认识

                b:应该充分认识到软件开发是一种组织良好，管理严密，各类人员协同配合，共同完成的工程项目

                c:使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好的技术和方法

                d:应该开发和使用更好的工具

1. 简述结构化范型和面向对象范型的要点，并分析他们的优缺点。

结构化范型要点：

1. 结构化范型也称生命周期方法学，属于传统方法学。

(2)传统的软件开发方法大部分采用瀑布模型。这种模型要求每一阶段都以前一阶段形成的文档为基础完成工作。

(3)每一阶段将要完成时，都要求开发人员进行验证或确认。

(4)瀑布模型要求在软件产品生成之前对用户需求进行尽可能精确的、完全的刻画。但要完成这种任务相当困难。

结构化范型优点：

1. 把软件生命周期划分成基干个阶段，每个阶段的任务相对独立，而且比较简单，便于不同人员分工协作，从而降低了整个软件开发过程的困难程度.
2. 在软件生命周期的每个阶段都采用科学的管理技术和良好的技术方法，且每个阶段都会进行严格审查，合格之后才进行下一阶段的工作，使得开发过程有条不紊的进行，保证了软件的质量，也提高了可维护性。

结构化范型缺点:

<1>当软件规模庞大时，或者对软件的需求是模糊的或会承受时间而变化时，开发出的软件往往不成功；而且维护起来仍然很困难。

面向对象范型四个要点：

1. 把对象作为融合了数据及在数据上的操作行为的统一的软件构件；

（2）所有对象划分成类；

（3）按照子类（派生类）和父类（基类）的关系，把若干个对象类组成一个层次结构的系统（类等级）；

（4）对象彼此之间仅能通过发送消息互相联系。

面向对象范型主要优点：

（1）按照人类习惯的思维方法，对软件开发过程所有阶段进行综合考虑；

（2）软件生存期各阶段所使用的方法、技术具有高度的连续性；

（3）软件开发各个阶段有机集成，有利于系统的稳定性；

（4）促进了软件的重用；

（5）降低了软件产品的复杂性，提高了软件产品的可理解性，简化了软件的开发和维护工作。

1. 假设自己是一家软件公司的总工程师，当把图给手下的软件工程师们观看，告诉他们及早发现并改正错误的重要性时，有人不同意这个观点，认为要求在错误进入软件之前就清除它们是不现实的，并举例说：“如果一个故障是编码错误造成的，那么，一个人怎么能在设计阶段清除它呢？”应该怎么反驳他？

答：在软件开发的不同阶段进行修改付出的代价是很不相同的,在早期引入变动,涉及的面较少,因而代价也比较低;在开发的中期,软件配置的许多成分已经完成,引入一个变动要对所有已完成的配置成分都做相应的修改,不仅工作量大,而且逻辑上也更复杂,因此付出的代价剧增;在软件“已经完成”是在引入变动,当然付出的代价更高。一个故障是代码错误造成的,有时这种错误是不可避免的,但要修改的成本是很小的,因为这不是整体构架的错误。