

从按下键盘到屏幕上显示字符，计算机经过了哪些过程？

(1) 键盘的构成，6 位的计数器经过两个八选一译码器对键盘进行扫描，在键盘没有任何按键时，扫描随着计数器的循环计数反复进行。(2) 若有键按下，扫描会发现该信号，该信号使得计数器停止计数，将扫描码发送给 8042。每一个键的位置都用一组数字表示，这组数字即是扫描码。(3) 8042 将该扫描码放在缓冲区中，并告诉 8259A 产生了中断。缓冲区大小为 1byte，如果此时键盘又有新的键按下，8042 将不再接收，一直等到缓冲区被清理。(4) 8259A 识别出键盘中断，将 CPU 的中断寄存器的特定位置为 1。(5) CPU 执行完一条指令，发现有中断存在，进入中断周期，通过 IDT 的映射将其处理交给键盘处理程序。

(6) 键盘处理程序第一步就是从 8042 的缓冲区将该扫描码移动到键盘缓冲区，此时可以接收下一个字符按键了。(7) 若键盘缓冲区非空，则开始处理这些按键码，首先将扫描码转换为 ASCII 码，如果是可打印字符，则将其显示，其他功能键做响应的处理。(8) BIOS 调用或者写显存实现，将该字符的 ASCII 码加上一些属性，放入显存中，字符发生器通过读这个特定位置的数据，将 ASCII 及其属性信息转化为一个光电矩阵，1 表示有光。字符串实际上是一个 ROM，用映射的功能实现这种转换，如 ASCII 码为 65，则字符发生器会将其转换为 a 的光点矩阵，颜色由电子发射枪控制。该光点矩阵可以作为显示器的输入信号。每个点都由三把电子发射器的光叠加而成。

键盘有一个键盘控制芯片，时时刻刻都在检测通过键盘的电流变化，如果有键盘按下，控制芯片会反馈给相应的中断程序，中断程序反应给 CPU，CPU 会把之前所做的一些任务停下，在晶振电路的控制下，通过相关的程序从内存中，或者 I/O 中读取数据来，然后在 CPU 的内部进行一系列的二进制计算，输入到内存或者 I/O 中，最后通过 GUI 来实现字符的显示。

Spring 的事务管理，事务是对一系列的数据库操作进行统一的提交或者回滚操作，如果插入成功，则一起成功，如果中间有一条出现异常，则回滚之前的所有操作。这样可以防止出现脏数据，防止数据库数据出现问题。Spring 中一般是使用 TransactionManager 进行管理，可以通过 Spring 的注入完成此功能。Spring 提供的事务管理可以分为编程式事务和声明式事务。编程式事务的代码量大，存在重复的代码比较多。Spring 的事务是通过声明式事务的方式对事务进行管理，即在配置文件中声明，通过 AOP 将事务切面切入程序，减少了代码量。

在 Java 中，可以作为 GC Roots 的对象：(1) 虚拟机栈中引用的对象 (2) 方法区中类静态属性引用的对象 (3) 方法区中常量引用的对象 (4) 本地方法栈中 JNI 引用的对象。

泛型：(1) 增加代码的安全性，编译时检查类型安全。(2) 增加代码的复用性，解决方法重载问题。(3) List<String>不能转为 List<Object>，因为泛型没有继承性，只是提供了限制功能。

代理模式的作用是为其他对象提供一种代理来控制对这个对象的访问。如果大量使用静

态代理模式会导致类的膨胀，如果事先不知道真实对象，则无法实现代理。装饰模式以对客户端透明的方式来扩展对象的功能，是继承关系的一个替代方案。继承是扩展类的功能，装饰模式是扩展对象的功能。装饰模式可以在不创建更多子类的情况下将对象的功能扩展。

装饰模式特点：（1）装饰对象和真实对象有相同的接口，客户端可以以和真实对象相同的方式和装饰对象交互。（2）装饰对象包含一个真实对象的引用。（3）装饰对象接收来自客户端的请求，把这些请求转发给真实的对象。（4）装饰对象可以在转发这些请求之前或者以后增加一些附加功能。

观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象，这个主题对象在状态上发生变化时，会通知所有的观察者对象，让其更新。

简单工厂模式是类的创建模式，又叫做静态工厂方法。简单工厂模式是指由一个工厂对象决定创建出哪一种产品类的实例，并且根据自变量的不同返回不同的实例。实质上是由一个工厂类根据传入的参数，动态的决定应该创建出哪一个产品类的实例。