1. ① 从DbC的角度看，Java方法声明中的throws子句反映了类（supplier）和其使用者（client）之间的怎样的权利/义务关系？

类和其使用者之间的权利/义务关系类似DbC中的服务提供者和客户。类的使用者有义务保证调用类时提供正确的条件，而一旦使用者提供了符合先验条件的输入参数，类保证运行得到符合后验条件的正确输出结果；反之如果使用者提供的参数不符合条件，从而导致运行过程中出错，类会抛出异常来提示使用者为何出错。

② Java子类若重定义父类中的方法，其throws的异常有何限制？

抛出的异常相当于对使用者未满足先验条件的描述。在DbC中，子类的先验条件应该比父类更弱（或相同，总之不能更强）。则子类重定义父类中的方法，可以抛出和父类完全相同的异常，抛出父类异常的子集（假如父类同时抛出好几个异常），抛出父类异常的子类型，不抛出任何异常，但就是不能抛出父类异常的父类型，或者父类没有的异常。

1. 解释checked exception, unchecked exception和error三者的定义以及使用的区别。

Checked exception: 程序与外界交互过程中出现的问题，在实际生产中依然存在，需要在程序开发（编写和调试）阶段对这类异常进行预测和相应的处理。

Unchecked exception: 程序内部/自身的问题，如一些违反契约的行为，或者bug。在测试阶段需要全部发现和解决，即实际发行版本中不应存在这类异常。

Error: 又system error，程序运行环境（如硬件设备和操作系统）可能存在的问题。无法仅通过程序代码提前预测和针对性解决，实际生产中仍可能存在且不可控。

Unchecked exception仅存在于开发阶段，在使用者看来，成熟的软件中这类异常一般可视为不存在（不可见）的；checked exception可能因为用户的不当操作出现，但软件会尝试解决或输出提示信息引导用户配合解决，避免程序直接崩溃。error会因为硬件或系统故障出现，此时程序仍可能直接崩溃，需要用户先解决系统故障再重新尝试运行。