1. **设计目标，节省编程时间，将可以自动完成的工作由程序完成**
   1. 自动生成数据库
   2. 字符类型的数据要能自动的提示长度
   3. 对于存储代码的数据要能自动转译存储和取出，同时要兼顾性能（可以考虑使用数据库维护，mapdb存储）
   4. 对数据关系的自动维护，可以自动展示一对多，一对一，多对一的数据，自动生成的方法名要符合逻辑并不能重复。需要载入相应对象的需要载入相应对象。
   5. 对需要执行特定sql语句的业务要能自动生成代码。
   6. 至少支持一种客户端框架，一种web端程序，一种数据库
2. **设计思路，使用javaweb+jquery+mapdb。**
   1. 支持数据库和实体类的对应关系。包括数据库表、字段、索引的自动生成。一个新应用对应一个数据库。
   2. 各类型的非必填属性可以设置、字符型数据按照长度进行限制，客户端要提醒。
   3. 对于存储代码的字段能够自动的关联对应的码表数据。一对一的两个表，一个实体类可以继承类外一个。
   4. 自动生成的对应关系的最好能够自动合并。关系类型参考hibernate配置。
   5. 客户端框架可以考虑jQueryui，web端使用sping+ibatis，数据库使用mysql+mapdb。关于mapdb，在只查看信息时，可以完全替代现有的数据库。
   6. 属性名称和数据库字段名称可以互相按照既定的规则生成，即只要其中之一不为空就可以自动生成代码。属性名->字段名：存在下划线“\_”则直接取小写形式，不存在则在尾部添加下划线。字段名->属性名：截取下划线“\_”，截取下划线后每个单词首字母大写（第一段除外）。
   7. 数据层次分为三层，从高到低分别为app(application),module,attribute(包括主键和外键)。属性中外键对应数据关系应该是外键所在的表只包含一对一和一对多的关系，自动生成方法名应该为getListBy[外键转属性]。
3. **数据库设计描述**
   1. 数据库表基本字段：id\_,name\_,created\_time
   2. config\_application应用表：每个应用对已一个数据库，一个系统。
   3. config\_object对象表：记录java对象和mysql表的对应关系，一条记录对应一个java类和一个mysql的表。
   4. config\_attribute属性表：可以记录基本数据类型的参数（paramter）也可以记录config\_objcet中的数据对应id以及码表对应的值。
   5. config\_unique唯一索引表：记录对应的config\_object中的唯一规则。
   6. 在进行叶面模板设计时，考虑到数据对象外键和修改时不能更改的属性时，在config\_attribute中添加记录字段用途属性，属性值包括（主键、外键、不能更改）。在叶面端设计到页面展示效果的问题，需要增加页面样式字段page\_style, 同时这个字段要能判断需要不需要转码处理。
4. **实践、经验、感想**
   1. 结合java和数据库特性设计本系统时，无法找到每个对象的具体定位，不能准确的说数据库对应类是一个java类的属性多一些还是mysql字段属性多一些，想起来一个问题，如果想要实现一个目标必须要有相应的工具和手段。所以设置应用的基础表为\_object(对象)和\_attribute(属性).
   2. 对象中添加固定的字段如主键。每个工程中同样要添加固定的支持页面的基本支持类及配置文件。
   3. 自动生成工具支持从数据库直接生成后，定型部署一个版本的自动生成工具
   4. 对模板文件进行管理，可以实现一套数据业务生成多种架构的代码。