

How TCB write Matlab Codes

前 言

Matlab 虽然拥有十分强大的功能，但并不是一个面向对象语言。因此在使用和开发的过程中需要建立一套合理的代码规范进行资源管理、提升代码可读性与降低耦合度。以下是自己使用过程中的一些经验教训。

关键词:Matlab; Coding Standard

目录

1	如何管理变量	1
1.1	命名规则	1
1.2	作用域管理	2
2	如何管理函数	2
2.1	命名规则	2
2.2	接口管理	2
2.3	函数调用	3
3	如何链接程序	3
3.1	迭代构建的情况	3
3.2	独立构建的情况	3

1 如何管理变量

1.1 命名规则

临时变量 驼峰法 [eg:tmpvValue; isChecked;]

全局变量 大写表示 [eg:MAXN; GENERATION;]

命名变量 下划线法 [eg:summer_data; origin_gragh;]

由于 Matlab 无法封装，所以在变量命名时尽量描述其 [所属的数据范围][阶段][属性名][用途] 等。

1.2 作用域管理

由于 Matlab 的变量在声明后在离开其声明的作用域时不会自己销毁，因此“尽可能局部地声明变量”这一铁律被打破了，不过仍然需要尽可能避免显式声明全局变量而是在文件中传递。相应地就必须对作用域做显式的管理。所以**一定要手动 clear 用到的临时变量！**尤其是在 for 之后

进行准确的作用域管理将带来一个巨大的优势——在程序运行结束后只需要将所有数据/新生成的数据均导出到 mat 中即可，视将采用的链接方式决定。

2 如何管理函数

2.1 命名规则

公用算法方法 帕斯卡法 [eg:Arima; GoalPrograming;]

项目算法方法 m_ 帕斯卡法 [eg:m_Arima; m_GoalPrograming;]

控制处理函数 首字母大写的下划线法 [eg:Calc_connection;]

2.2 接口管理

由于 Matlab 无法重载，只能通过可变参数列表的方式形成比较有限的多态，因此在函数接口设计时需要尽可能地将参数列表设计为一致的。好在 Matlab 的参数列表还是比较

现代的，传递比较方便。

2.3 函数调用

由于 Matlab 的函数参数传递是值传递，因此在函数内部对参数的修改不会影响到函数外部的变量，因此在函数内部对参数的修改需要通过返回值的方式进行。

Matlab 调用函数文档及其方便，所以不要偷懒给每一个函数都写上必要的说明，尤其是算法函数提供较细致的接口说明

3 如何链接程序

3.1 迭代构建的情况

mat 中使用同样的文件名，这样调用不同阶段的数据文件就可以查看相应情况的结果。

数据处理最好使用迭代构建。比如 [读取 → 清洗 → 预处理 → 分类]

添加属性最好使用流程构建。比如已经有了若干结论，在此基础上需要添加新的属性，这时候就需要将原有的数据文件作为输入，添加新的属性后输出到新的文件中。

3.2 独立构建的情况

在不同线性的处理流合并、分析等都最好使用独立构建。包含若干需要的中间数据，只将得到的少量所求数据放入文件，因为往往这是一个阶段的结果，也常常是需要在此基础上进行下一步的处理的起点。