

新员工编码实训计划

第一天

一. 编码风格讲解(2h)

主讲：zbc

《编码风格》主要目的：

1. 掌握公司编码规范的基本要求；
2. 了解编码规范希望达成的背后目的，即，写出具有良好风格的代码；
3. 了解什么样的代码是风格良好的代码，这样写有什么好处；
4. 了解编码六步法的基本操作方式，六步法中每一步的基本目的；

二. 训练课题要求讲解(1h)

讲解训练课题的大概情况，训练需要完成的任务，达到的目标。

三. API设计(4h)

[任务]

- 安装必要工具，构建开发环境(参考tools-intro.md的说明)
- 把单元测试，覆盖率，valgrind检测全部运行一遍，产出预期的运行结果
- 完善demo，该demo可构建readme.md中的范例JSON
- 思考json.c中带"想想"标记的问题，给出答案
- 把json.h中的API作为一个初版，根据demo.c进行分析，看有什么可以改进的地方，以及各API实现有什么要注意的地方；

[方法]

API设计，以及对API进行分析改进的方法：

- 第一步. 先分析场景，梳理设计需求（设计要注意的要点）；
- 第二步. 给出基本方案，方案列出有哪些API，并写出API的原型，和功能描述；
- 第三步. 有了基本方案后，切换为用户视角，利用基本方案中描述的API完成demo；
- 第四步. 根据demo反复思考，有哪些细节需要考虑，有没有让用户写代码更简单的实现方法；
- 第五步. 修改API方案，将构思成熟的API输出到表格

[产出]

1. 思路分析思维导图

- 场景分析
- 功能分解
- 注意事项

2. demo代码

切换为API使用者的视角，利用API去实现readme.md中要求的功能，即生成一个复杂json值的内存数据结构表示

3. API设计表格

表格包括至少这四列：API原型，API功能，API参数和返回值说明，设计思路（为什么这么设计，以及后续实现思路）

第二天

一. 第一天练习结果点评(1h+...)

第一个小时，讲师抽取部分学员的练习结果，进行集中点评。然后大家分头做接下来的练习，同时，讲师单独和剩下的学员一一沟通练习问题；

二. json-data-structure实现(6h)

1. 单元测试案例编写

2. API实现

3. 调试运行

给出单元测试运行结果(运行输出的文本或屏幕快照)

4. 工具检查(lcov, valgrind)

给出覆盖率报告(web页面，或gcov文件)，给出valgrind运行结果(运行输出的文本或屏幕快照)。

第三天

一. json-data-structure实现结果点评(1h)

二. json_save实现

1. 单元测试案例编写

2. API实现

3. 调试运行

4. 工具检查(lcov, valgrind)

第四天

一. 分组相互review

分为两组，相互review第3天的练习成果。发现问题更多的团队获胜，奖励小礼物。

review结束后，大家将review后的代码打包发给辅导员，命名为 "<检视人姓名>review<被检视代码的作者姓名>.zip"

[规则]

1. 发现问题在代码处标记，如：`//zbc: 未检查参数有效性`
2. 所有违反checklist的都是问题, 如命名，缩进，注释，参数检查，返回值检查等
3. 命名，缩进，注释等只影响代码阅读的问题，同类的只记一个问题；
4. 会导致运行bug，或有bug隐患的代码，每个地方都记一个bug，比如返回值不检查，内存泄漏，内存越界访问，数据未初始化就使用等。

二. 查缺补漏，修正json-data-structure和json_save存在的问题