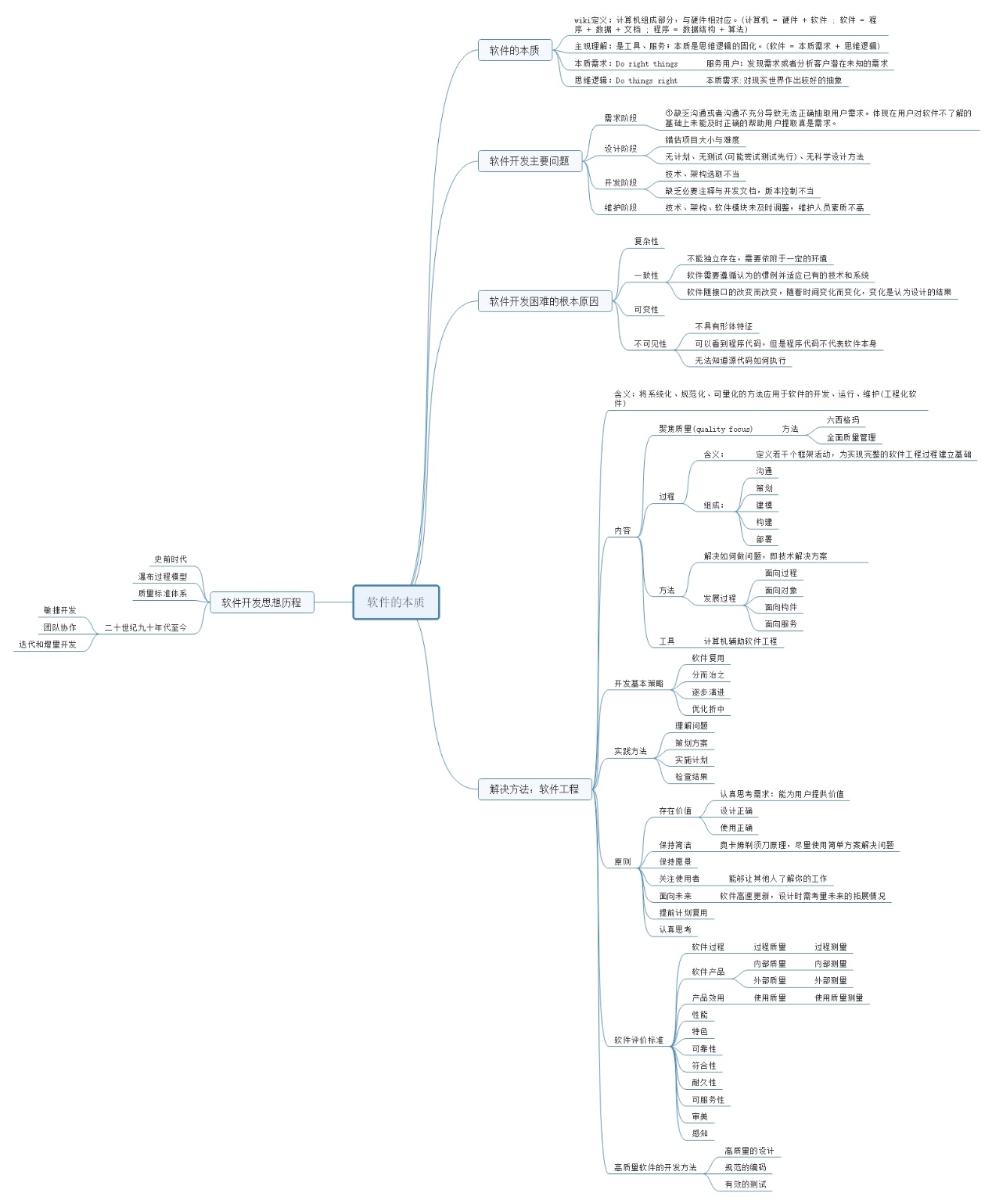
### Week 1

一. 学习笔记

1.软件的本质

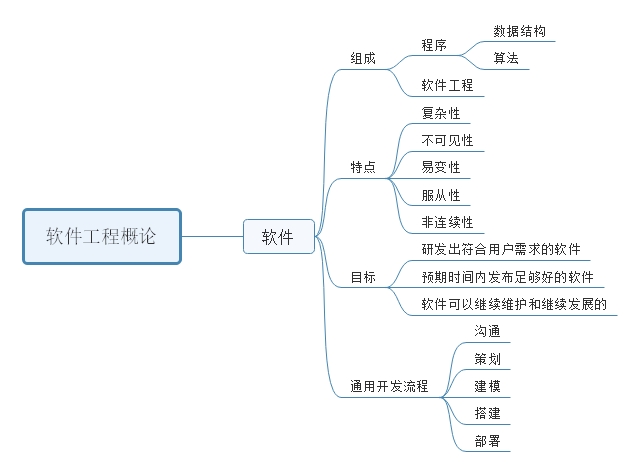


2. 作业程序 见 main.py

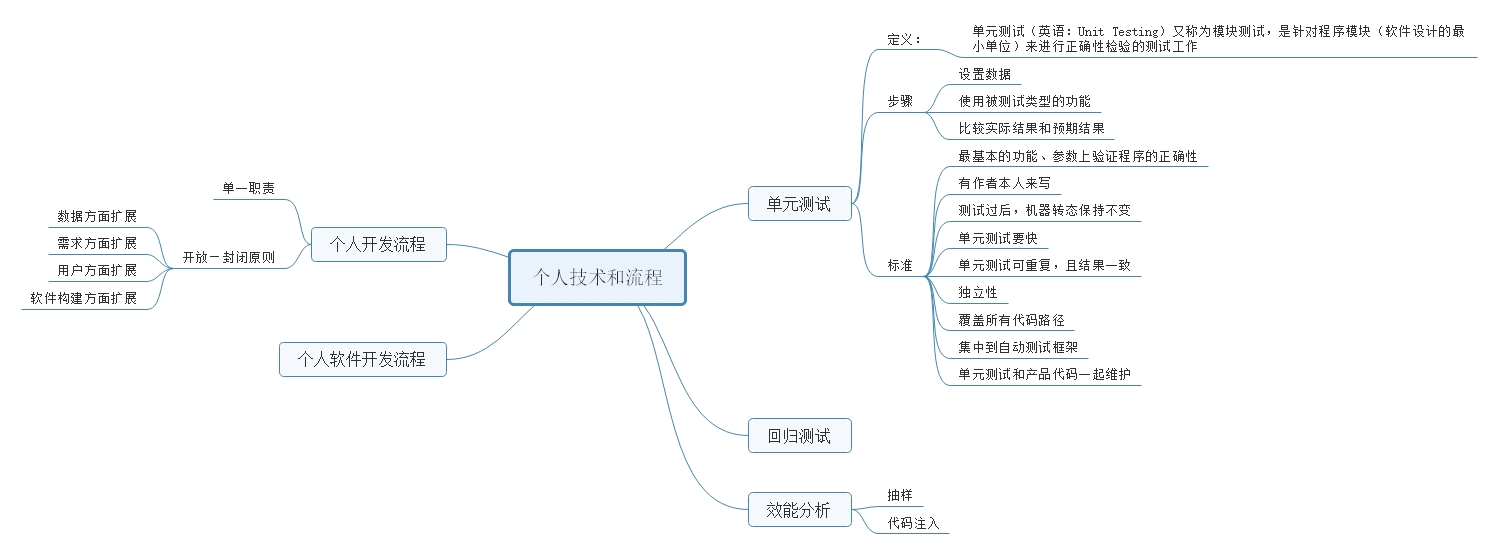
### Week 2

一、学习笔记(构建之法学习笔记)

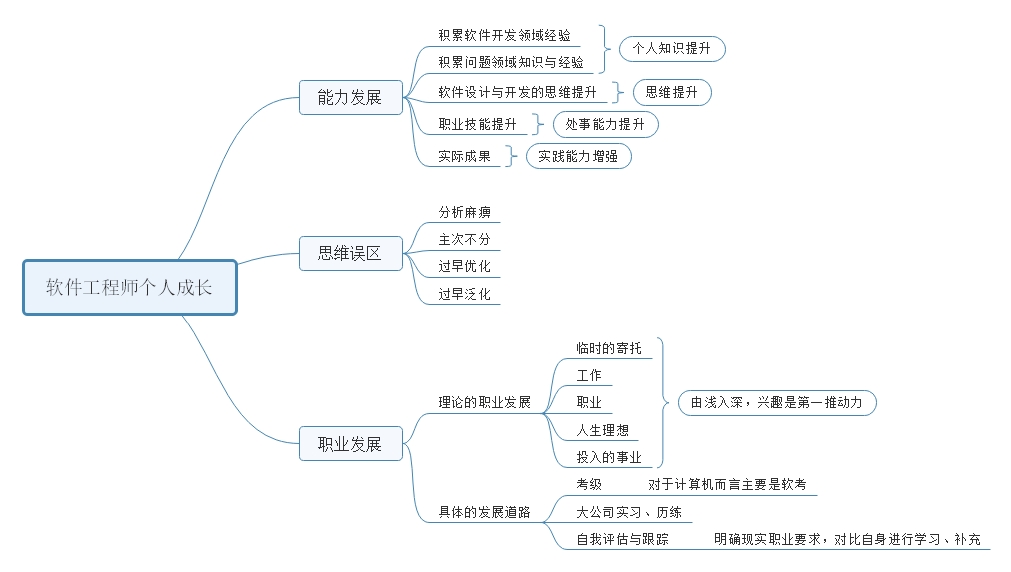
1、软件工程概述。



2、软件测试与能效分析。



3、个人发展。



4、 个人感悟。软件工程是为了更好的处理软件开发中的各种问题，鉴于问题的多样性，应该针对不同的软件问题寻求不同的解决方法。瀑布开发不一定不能使用，敏捷开发不一定全能，针对不同的需求、不同的现实环境，不同的开发人员，应该寻求不同的解决方案。技术只是手段，目标才是根本。软件开发立足于现实，服务于理论；理论指导开发，同时受开发影响。

5、个人项目总结。

个人任务的技术重点在于表达式的计算。常规思路有两种：①借助数据结构栈的概念实现。借助两栈分存数据与运算符，依据运算符优先级进行计算、存储，具体实现如github示例。②使用分治思想将任务分解为加减、乘除；使用递归思想分别计算加减、乘除。具体实现如<https://blog.csdn.net/qq_25736745/article/details/88320095>所示。

二、项目实践。

见文件Main.java。

三、问题。

Github上 java 起始项目示例存在问题。

1、不能产生除法

2、除法运算存在问题。

解决方法：

1.1 修改运算符取值范围。

2.1输入数据之前对数据进行控制，保证能够整除。

2.2除法运算时，修改被除数=被除数-余数。

四、附录(PSP记录)

PSP表格1 时间记录表格（单位：分钟）

| **PSP2.1** | **Personal Software Process Stages** | **预估耗时（m）** | **实际耗时（m）** |
| --- | --- | --- | --- |
| Planning | 计划 | 1h | 30min |
| · Estimate | · 估计这个任务需要多少时间 | 30min | 20min |
| Development | 开发 | 1h | 30min |
| · Analysis | · 需求分析 (包括学习新技术) | 30min | 10min |
| · Design Spec | · 生成设计文档 | 30min | 10min |
| · Design Review | · 设计复审 (和同事审核设计文档) | 1h | 10min |
| · Coding Standard | · 代码规范 (为目前的开发制定合适的规范) | 10min | 10min |
| · Design | · 具体设计 | 30min | 40min |
| · Coding | · 具体编码 | 30min | 20min |
| · Code Review | · 代码复审 | 10min | 20min |
| · Test | · 测试（自我测试，修改代码，提交修改） | 30min | 40min |
| Reporting | 报告 | 10min | 10min |
| · Test Report | · 测试报告 | 10min | 8min |
| · Size Measurement | · 计算工作量 | 20min | 22min |
| · Postmortem & Process Improvement Plan | · 事后总结, 并提出过程改进计划 | 30min | 20min |
|  | 合计 | 7h | 5h |

说明：

* 实现之前先在PSP中预估时间
* 实施后各个环节实际花费多少时间也请做记录
* 表中有一项： Estimate 指的“预估”这个活动，“预估时间”也是一项任务。

例如：我估计自己需要花30分钟来估算出整个项目需要多少时间完成，结果我花了20分钟估算出整个项目需要6个小时完成。Estimate这一项应该在“预估耗时”填写30分钟，实际耗时填写“20”分钟。

* 一级和二级活动的包含关系：

Planning 这个一级活动包含了1个二级活动（Estimate），Development 这个一级活动包含了8个二级活动，Reporting 这个一级活动包含了3个二级活动，二级活动时间累加即为一级活动时间。

### Week 3

一、项目实践与测试

<https://github.com/DoraemonSCL/Calculator/tree/java/src>

二、缺陷记录日志

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **缺陷记录日志** | | | | | | | |  |
| **日期** | **3/15** | | | |  |  | | |  |
| **程序号** | **1** | | | |  |  | | |  |
| **日 期** | **编号** | **类型** | **引入阶段** | **排除阶段** | **修复时间** | **相关缺陷** |  |
| **3/15** | **10** | **文档** | **执行** | **查阅资料** | **3min** |  |  |
|  | **注释含有中文，windows10命令行执行报错** | | | | | |  |
| **3/16** | **80** | **函数** | **编码** | **查阅资料** | **20min** |  |  |
|  | **无法产生除法** | | | | | |  |
| **3/16** | **80** | **函数** | **调试** | **编译、查询资料** | **30min** |  |  |
|  | **除法计算出错，只求商，未考虑余数问题。** | | | | | |  |
|  |  |  |  |  |  |  |  |
|  |  | | | | | |  |
|  |  |  |  |  |  |  |  |
|  |  | | | | | |  |
|  |  |  |  |  |  |  |  |
|  |  | | | | | |  |

缺陷类型标准

|  |  |  |
| --- | --- | --- |
| 类型编号 | 类型名称 | 描    述 |
| 10 | 文档 | 注释，消息 |
| 20 | 语法 | 拼写，标点符号，打字，指令格式 |
| 30 | 联编打包 | 变更管理，库，版本控制 |
| 40 | 赋值 | 说明，重名，作用域，限制 |
| 50 | 接口 | 过程调用和引用，输入/输出，用户格式 |
| 60 | 检查 | 出错信息，不合适的检查 |
| 70 | 数据 | 结构，内容 |
| 80 | 函数 | 逻辑，指针，循环，递归，计算，函数缺陷 |
| 90 | 系统 | 配置，记时，内存 |
| 100 | 环境 | 设计，编译，测试，其它支持系统问题 |
|  |  |  |

三、项目拓展

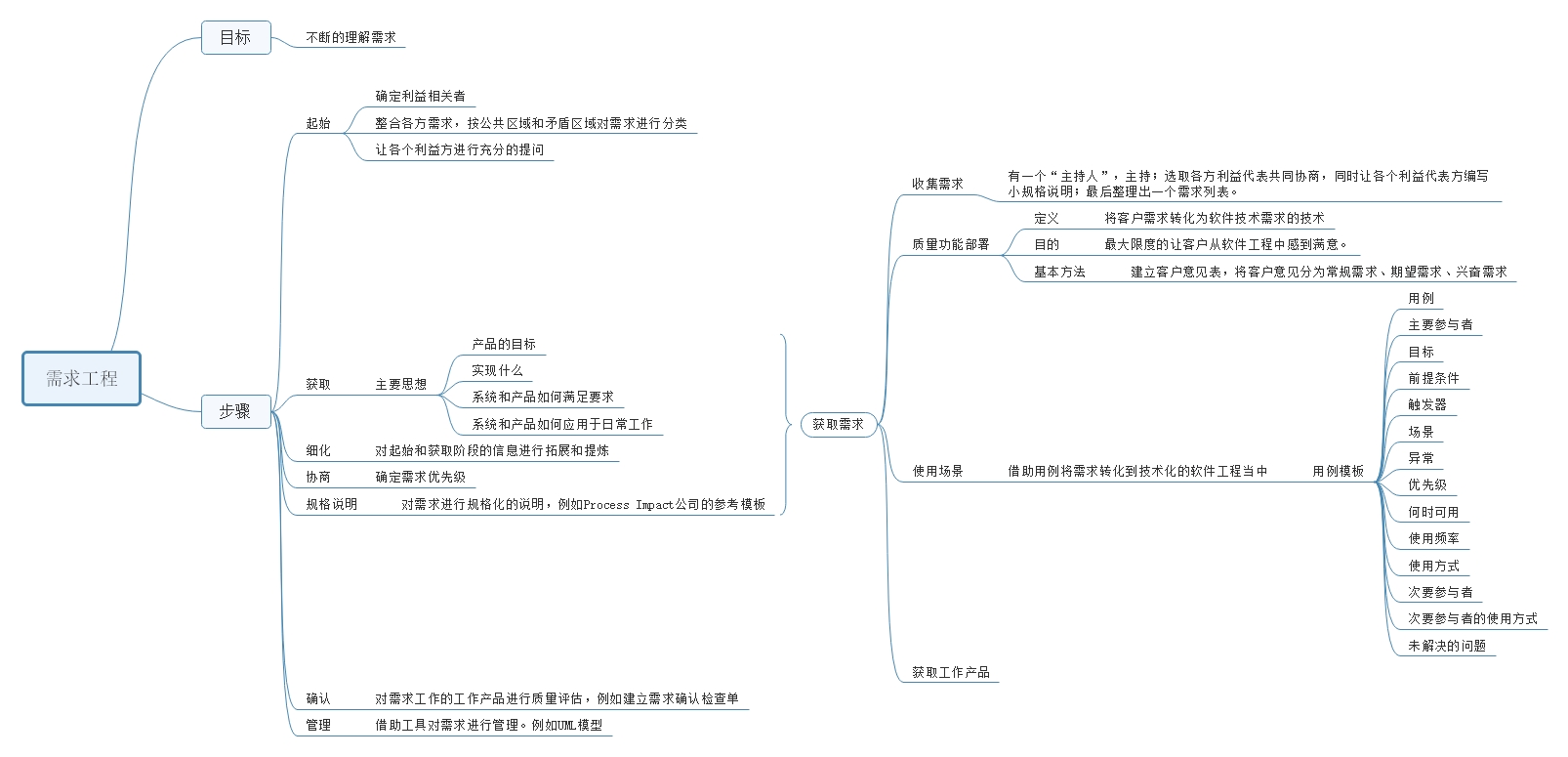
<https://blog.csdn.net/qq_25736745/article/details/88320095>

四、备注

最近事情较多，项目拓展未进行重复编码，待空闲时将项目进行整合以博客发出。

### Week 4

一、学习笔记



二、思考

1.《构建》ch10中的典型用户与用例模型中的参与者有什么关系？

区别：

1)典型用户：强调典型的软件使用者，从具体使用者的角度进行考虑。

2)用例模型的参与者：强调参与软件开发的事物，包括人或者其他事物。从应用开发的角度进行考虑。

2. 《构建》ch10中的典型场景与用例模型中的用例描述有什么关系？

区别：

1)典型场景：描述达到目标必须经历的过程。

2)用例描述：说明用户使用用例完成的任务。

三、总结

个人理解的工程化获取需求的基本方式：

1)明确利益用户，协商利益群体的各种需求，按照优先级的方式对需求排序；

2)借助用例从技术角度对需求进行描述，填写、制定用例模板并进行确认。

3)在实际使用过程中，小周期迭代，按优先级进行处理，及时更新需求。