

OOP大作业

一.摘要

原有OOP大作业存在着一些问题：

1. 主观成分较多，强调功能，不强调项目设计，与授课内容有一定偏离。
2. 扩展项逐年变难，不仅加重了同学负担，还只收获了一些与课程无关的知识。
3. 只为能写出更多的功能，忽视了程序的可拓展性，导致同学缺少计算机系后续课程所需的设计能力。
4. 抱大腿情况严重，很多时候能力强的同学写完整个项目，能力较弱的同学没有获得锻炼。

新的大作业希望达到的是：

1. 中等难度，对一般学生略有挑战性，但不至于不能完成。
2. 加入较为客观评价的评价方式。
3. 加强设计方面的训练。让同学们能够体验和反思框架的设计过程，并体验代码再开发的过程，为软件工程课打下基础。

二.大作业流程

1. 第一阶段：需求设计，快速开发阶段

此时老师会公布项目题目和第一阶段基础需求。每组同学应该按照题目要求，设计一个合理、符合OOP思想、具有拓展性的框架，并实现基础需求。

第一阶段结束时，你应该提交自己的代码、设计文档和实验报告，在实验报告中展示或测试自己代码的例子。

注意：第一阶段应主要按照需求来完成设计，不用实现太多的拓展内容，因为这些内容有可能是第二阶段的基础需求。

2. 第二阶段：代码互评，再次开发

第二阶段开始后，老师会公布第二阶段基础需求和扩展需求，同时展开第一阶段代码的互评。每组同学会收到除自己外，若干组（匿名）第一阶段的代码。

首先，你应该阅读下发的代码，联系第二阶段需求，分析代码是否易于进一步开发。你需要对这些代码进行排序，并简要分析每份代码的优劣。我们会将排序的投票结果作为互评分数。

接下来，你有两个选项：要么基于你排序第一名的代码进行进一步开发，要么你可以选择仍然在自己的代码上进行开发（在这种情况下，你必须详细说明自己的代码为何比其他组代码都好，并且这一阶段分数将会乘上一个小于1的衰减因子）。

然后你应该在选定的代码基础上完成基础需求，选做拓展需求。为了模拟真实应用场景，你不得改动第一阶段的**测试代码**（因为在真实项目中，修改接口往往会牵一发而动全身，容易浪费大量时间），在其基础上进行拓展。如果实在需要改动，会按照改动幅度酌情扣分。

第二阶段结束时，你应当提交自己拓展后的代码、拓展后的设计文档和展示自己代码功能的例子，并且保证原有测试代码能够正常运行。

注意：提交的代码分为库和测试代码，完成第二阶段时，库可以适当添加、修改，但原接口需要兼容，即第一阶段的测试代码仍然要能运行。除此之外，第二阶段还需要再提交一份测试代码，用于测试第二阶段的需求。

三.分数说明

得分构成如下：

- 1. 第一阶段的功能是否满足基础需求。（占30%）
- 2. 第二阶段你对其他组的评价会由助教判定是否合理。（占5%）
- 3. 第二阶段他人对你的排序会作为投票，所有组按得票排序，得分顺次递减。这个得分会作为你第一阶段的设计得分。并且你会收到来自其他组对你框架的评价（匿名）。如果你对对方的评价有意见，可以写出反对意见，言之有理可加分。（占10%）
- 4. 第二阶段的功能按完成基础需求和拓展需求进行打分。（基础需求占25%，拓展需求为bonus）
- 5. 第二阶段会由助教按照课堂展示、设计文档和代码给出设计分数。（占15%）

注意：

- 1. 若使用自己第一阶段代码，4、5项得分会乘上衰减因子（暂定0.8）；
- 2. 大作业的总分不会超过120%。

四.时间节点

时间	说明
第5周周日(3.31)	提交组队和选题信息
第6周周一(4.1)	第一阶段开始
第10周周日(5.5)	第一阶段任务提交
第11周周五(5.10)	完成互评，第二阶段开始
第16周(待定)	展示
第17周周日(6.23)	第二阶段任务提交