

哈尔滨工业大学

计算机科学与技术学院/国家示范性软件学院

2020 年秋季学期

《软件过程与工具》课程

实验报告

Lab 1: 增量模型应用+结对编程实践

姓名	学号	联系方式
蔡建华	1183710119	13599865378
尹大通	1183710103	19845177857

目 录

1 项目概述.....	1
1.1 总体需求分析.....	1
1.2 软件系统总体设计.....	2
1.3 开发任务计划.....	4
1.4 开发环境与工具.....	4
2 第 1 轮迭代：基本系统.....	4
2.1 迭代目标.....	4
2.2 需求分析.....	5
2.3 设计方案.....	5
2.4 结对编程过程记录.....	6
2.5 单元测试.....	7
2.6 集成测试.....	8
3 第 2 轮迭代：增量 1.....	8
3.1 迭代目标.....	8
3.2 需求分析.....	9
3.3 设计方案.....	9
3.4 结对编程过程记录.....	11
3.5 单元测试.....	12
3.6 集成测试.....	13
4 第 3 轮迭代：增量 2.....	13
4.1 迭代目标.....	13
4.2 需求分析.....	13
4.3 设计方案.....	14
4.4 结对编程过程记录.....	17
4.5 单元测试.....	18
4.6 集成测试.....	19
5 项目总结.....	19
5.1 增量模型应用总结.....	19
5.2 结对编程总结.....	20

1 项目概述

(1) 项目名称: 100 以内加减法练习小软件

(2) 项目简要介绍:

本项目为 100 内加减法的小软件, 提供 100 以内的加减练习算式, 面向的使用用户群体为小学一年级学生。软件可自动生成 100 内加减练习题, 能够进行答案校验同时可以换题, 并保证一次练习过程中题目不重复。此外, 软件还提供了音乐、错题集、计时、正确(错误)作答题数记录等功能, 界面友好, 简洁易操作。

1.1 总体需求分析

(1) 项目要开发的软件的目的和意义(回答为什么要开发该项目)。

由于老师教学任务的需要, 同时为了提高学生的数学运算能力, 为学生后面的学习打下基础, 需要一款软件能够提供难度有限的加减法算式供学生使用。该加减法练习小软件可以作为老师的作业, 同时也能分担家长教育孩子的压力。

作为小一年级学生练习简单加减法使用的工具, 软件能自动生成题目, 供学生练习加减法, 并进行答案校验, 还设置有错题集, 各方面提高学生数学运算能力。

软件界面友好, 简洁易操作, 难度符合小学一年级学生的水平, 满足小学一年级学生学习的要求, 有利于激发学生学习热情。

(2) 软件系统的涉众分析(列出该系统跟哪些相关人员有关, 其关心的核心内容)。

小学一年级学生: 能够利用该软件得到大量的练习, 能够进行答案校验, 软件自动记录练习时间, 做错题目会进入错题集, 便于练习后进行反思总结。软件中还提供音乐功能, 界面简洁、易操作。

老师: 该软件的练习题可以作为老师的作业来执行, 减轻了老师出题的压力。

家长: 利用该软件, 家长可以更有效地辅导孩子学习, 掌握孩子学习情况, 同时减轻了家长教育孩子的压力。

(3) 软件系统的功能需求: 可从用户视角来描述功能需求(即用户故事)。

学生：需要随机生成各种算式，但是题目不能太难。要能够进行答案校验，做错的题目放入错题集中，以便我后续查看。界面要简洁，操作不能太麻烦。如果能边做题目边听歌那就更好了。

老师：题目难度要适当，题目也不应该重复，最好来个计时功能，同时记录正答题数、错答题数。

家长：练习时间要记录，练习题应该要有很多，还不能重复，做错的题目也要放进错题集。练习时间要记录，方便我监督跟踪孩子的学习。

(4) 软件系统的非功能需求：包括性能要求、界面要求、技术指标等。

要求生成题目、校对答案的过程足够快；

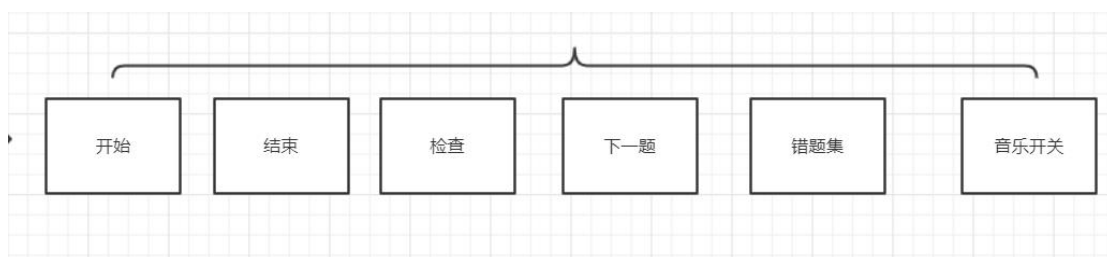
界面简洁美观，操作简单；

题目随机生成但保证题目不重复且难度适合小学一年级学。

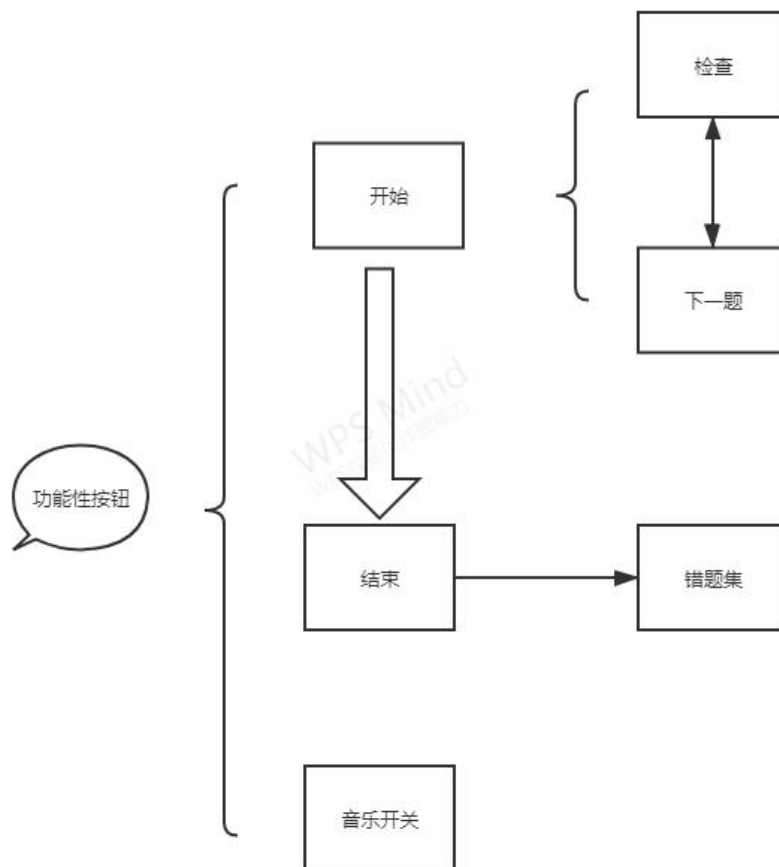
1.2 软件系统总体设计

(1) 软件系统功能结构设计方案：给出软件功能模块及模块间的关联关系。

该软件除了提供基本的生成练习题、检查答案、换题等功能外，还提供了计时、错题集记录、背景音乐等功能。界面中包括“开始”、“结束”、“检查”、“下一题”、“错题集”功能按钮。功能模块如下图：

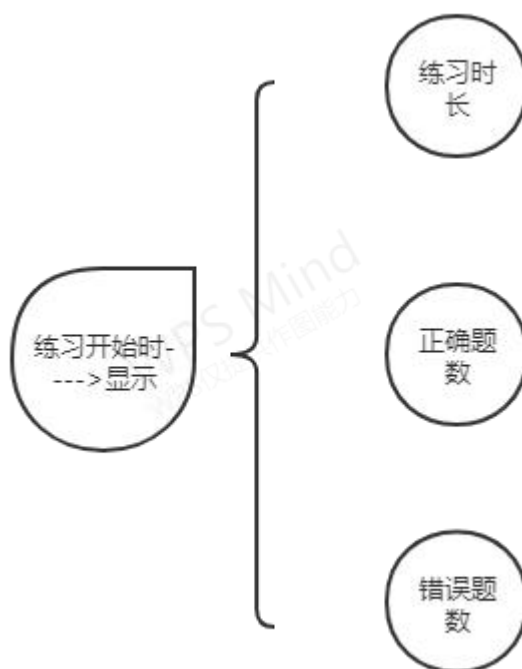


当点击“开始”按钮时开始做习题，同时计时开启。在输入框中输入作答后可以点击“检查”按钮进行答案校验，软件会给出作答正确或者错误的提示，若作答错误，题目会加入到错题集中；点击“下一题”会换题，同时自动评判答案并将错题加入“错题集”中；点击“结束”，结束练习，停止计时；这时，再点击“错题集”可以查看答错的题目。



功能性按钮模块间关系如上图所示。

除此之外，软件还会记录正确题数、错误题数，并显示实时练习时长。



(2)软件系统相关的数据结构设计: 给出必要的数据结构设计, 包括配置参数、缓冲数据结构等

练习式: 加减法式子中的加数或被减数或减数随机生成, 具体由 python 中的 random.randint 随机生成。整个式子以字符串的形式保存在 python 的数组中, 之后再显示在界面上。

错题集: 记录作答错误的式子, 式子同样以字符串形式保存在 python 的数组中。查看时再打印。

正确(错误)作答数: 设置一个变量, 初始值为 0, 作答正确(错误)则加 1。

除此之外, 程序中用到了其它的变量, 总体数据结构并不复杂, 主要用的有上述的数组结构。

1.3 开发任务计划

迭代过程的划分, 各迭代计划完成的增量任务, 进度安排。

(1)第 1 轮迭代的任务:

提供 100 以内加减法练习题, 能自由换题, 提供答案检查。

(2)第 2 轮迭代的任务:

保证题目不重复, 记录正确数量、错误数量, 并能够提供计时功能。

(3)第 3 轮迭代的任务:

增加错题集; 提高用户的使用体验, 增加音乐, 美化界面。

1.4 开发环境与工具

开发语言: python (python 版本号为 3.8)

开发环境: Windows10

开发工具: PyCharm 2020

2 第 1 轮迭代: 基本系统

2.1 迭代目标

简要介绍本轮迭代的增量任务目标, 所需完成的任务。

- (1) 任务 1: 生成 100 以内的加减练习题（减式中被减数不能小于减数）。
- (2) 任务 2: 检查输入答案是否正确
- (3) 任务 3: 换题。答完一道题后转到下一题。
- (4) 任务 4: 构造初始的 UI 界面。显示题目，提供‘开始’、‘结束’、‘检查’、‘下一题’等功能按钮。

2.2 需求分析

提供了基本的功能：自动生成练习题、开始计时、检查、换题、结束，实现了基本的 UI 界面，这些功能已经能够满足小学一年级学生的练习需求了；满足老师要求的习题难度；同时能够为家长分担教育孩子的压力。

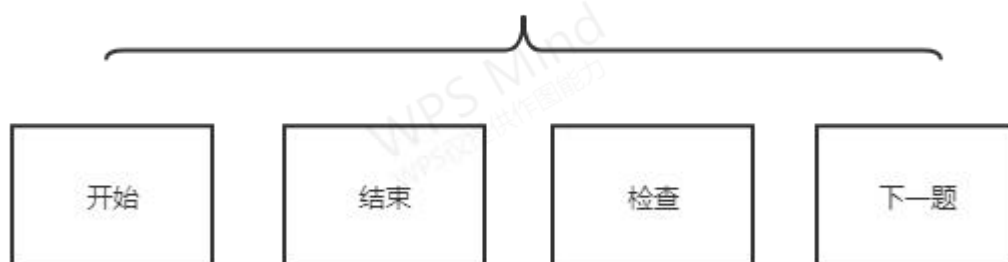
界面简洁，易操作，作答输入方便。

2.3 设计方案

针对本轮迭代的开发内容给出设计方案：

- (1)详细设计方案：模块功能结构设计，界面原型设计等。

设计了“开始”、“结束”、“检查”、“下一题”等几个功能按钮。点击“开始”后即可开始练习，练习过程中可以点击“检查”进行答案检验，点击“下一题”进行换题。点击“结束”，结束练习。功能模块如下图：



界面原型如下图：

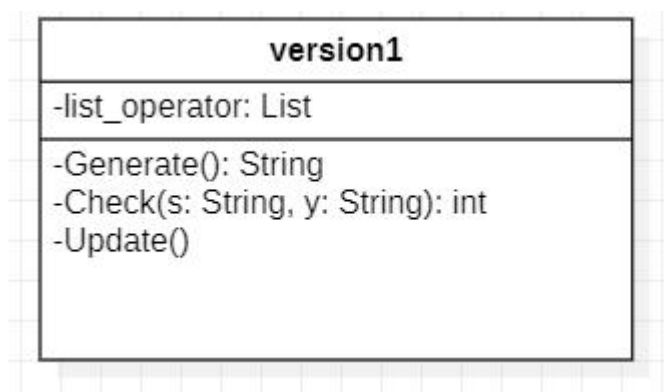
100以内加减法练习

91 - 79 =

回答正确✓

开始
结束
检查
下一题

(2)模块相关的设计类（Design Class）：给出每个类的名字、属性、方法。
类中主要有 3 个方法，执行生成算式、检查、换题等功能。类图如下：



(3)所需的数据结构。

字符串：暂时保存练习式，用于后续的答案校验；

list 数组：保存两个常量‘+’和‘-’，用于随机生成加减式。

2.4 结对编程过程记录

(1) 角色切换与任务分工

表 1-1 第 1 轮迭代过程结对编程角色与任务分工

日期	时间(HH:MM - HH:MM)	驾驶员角色	领航员角色	本段时间的任务
10.17	16:00-17:00	蔡建华	尹大通	练习式随机生成
10.17	17:00-18:00	尹大通	蔡建华	函数“检查答案”、“更新”
10.18	19:00-20:00	蔡建华	尹大通	写前端界面

日期	时间(HH:MM - HH:MM)	驾驶员角色	领航员角色	本段时间的任务
10.18	20:20-20:50	尹大通	蔡建华	修改前端界面的一些问题

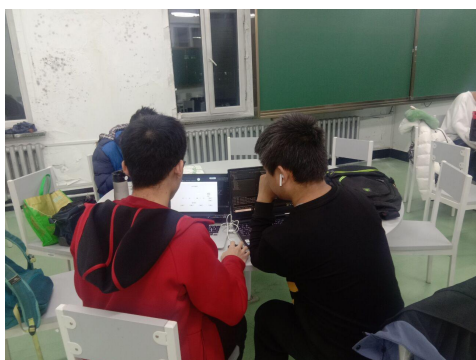
(2) 工作日志

由领航员负责记录，记录结对编程期间的遇到的问题、两人如何通过交流合作解决每个问题的。

表 1-2 第 1 轮迭代过程结对编程工作日志

日期/时间	问题描述	最终解决方法	交流过程
10.17/16:40 -16:50	生成的减式中被减数小于减数	生成时若是减式则判断被减数与减数的大小	这是个疏忽，很简单的问题，所以我们马上想到只需在生成时加个判断即可，若是被减数小于减数，则再生成一个新的式子。
10.17/17:50 -18:00	如何判断输入的答案是否正确	使用 python 中的 eval 方法可以直接计算已经保存成字符串形式的算式的答案，接着便可进行比较	由于生成的算式是以字符串形式保存的，因此计算正确答案需要做一个决策。后来我们发现 Python 中提供了一个方法 eval 可以直接计算字符串形式的算式，于是这个问题便解决了。
10.18/20:30 -20:38	界面一些按钮显示不了	重新设置了按钮布局	在 python 中，我们调用了 tkinter 库来写 UI 界面，由于编程过程中几个按钮的位置有一点冲突，导致了显示的问题。问题比较小，但藏得又比较深。

(3) 结对编程工作现场照片



第 1 次迭代结对编程现场照片 1



第 1 次迭代结对编程现场照片 2

2.5 单元测试

对本轮迭代所开发的每个类进行单元测试。

表 1-3 第 1 轮迭代过程的单元测试记录

日期/时间	类名: 方法名	输入数据	输出数据	结论
10.17/16:50 -17:00	version1:generate	无	68 - 15	正常
10.17/16:50 -17:00	version1:generate	无	20 - 33	不正常。 被减数小于减数, 需修改。
10.17/16:50 -17:00	version1:generate	无	55 + 22	正常
10.17/16:50 -17:00	version1:generate	无	30 - 8	正常
10.17/17:50 -18:00	version1:check	"20 + 15", "35"	1	正常
10.17/17:50 -18:00	version1:check	"20 + 15", "11"	0	正常
10.17/17:50 -18:00	version1:check	"19 - 7", "12"	1	正常
10.17/17:50 -18:00	version1:check	"19 - 7", "2"	0	正常
10.17/17:50 -18:00	version1:update	无	"83 - 27" (新生成的算式)	正常

2.6 集成测试

表 1-4 第 1 轮迭代过程的集成测试记录

日期/时间	功能名称	测试用例	测试结果描述	结论
10.18/21:00 -21:30	开始	点击“开始”按钮	顺利开始练习	正常
10.18/21:00 -21:30	检查	点击“检查”按钮	正确校验了答案	正常
10.18/21:00 -21:30	下一题	点击“下一题”按钮	顺利换题	正常
10.18/21:00 -21:30	结束	点击“结束”按钮	结束了练习	正常

3 第 2 轮迭代: 增量 1

3.1 迭代目标

简要介绍本轮迭代的增量任务目标, 所需完成的任务。

- (1) 任务 1: 实现正确作答题数的记录。
- (2) 任务 2: 实现错误作答题数的记录。

(3) 任务 3: 计时功能。

(4) 任务 4: 避免题目重复。

3.2 需求分析

在学生的练习过程中, 应该避免出现重复的题目, 而 100 以内的加减法在一定的练习量下还是有重题的可能, 因此此次迭代在代码中实现了题目不重复的内置要求。

计时功能可以帮助学生记录自己的练习时间, 观察自己的做题速度, 并可以根据此合理调节自己的答题速度。

记录正确题数、错误题数可以帮助学生了解自己的答题情况, 同时也可以作为家长了解孩子答题、学习情况的根据。

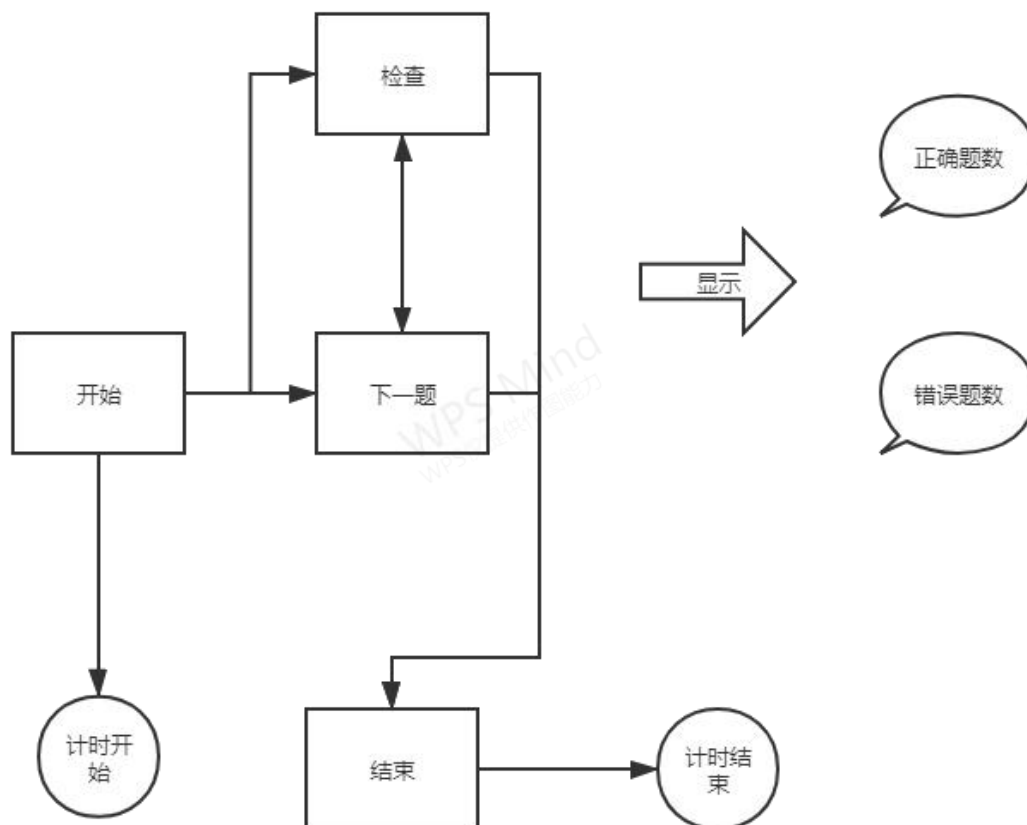
3.3 设计方案

针对本轮迭代的开发内容给出设计方案:

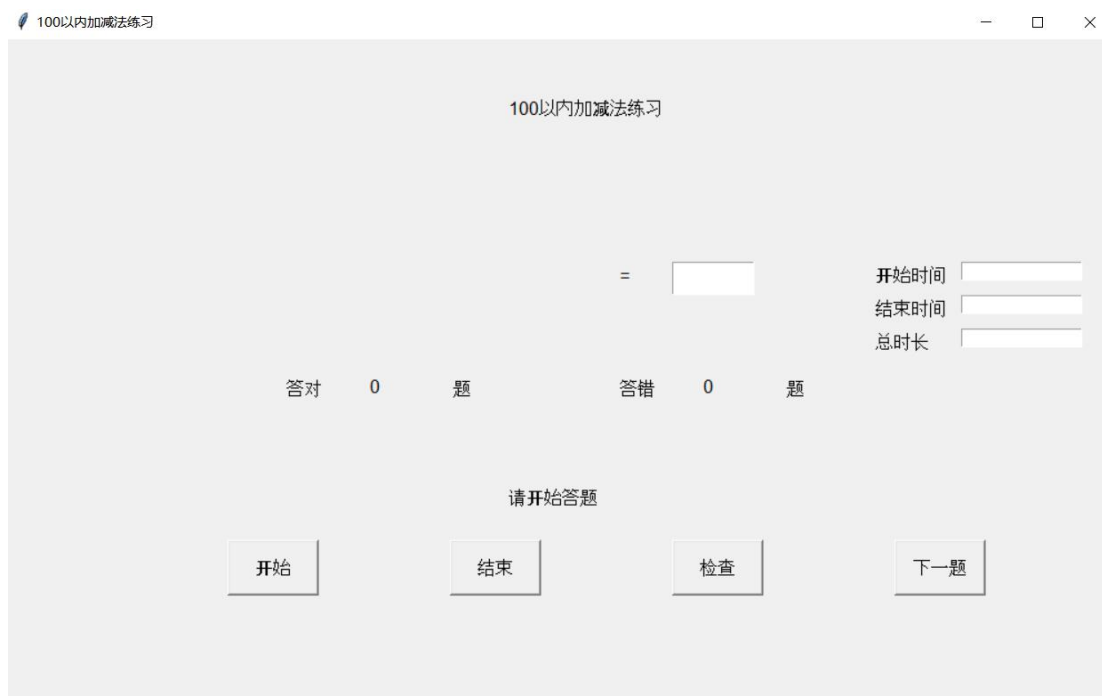
(1)详细设计方案: 模块功能结构设计, 界面原型设计等。

新增正确作答题数、错误作答题数的记录, 并实现计时功能(在练习开始时自动开始计时)。同时, 代码确保了了一次练习过程中题目不重复(这相当于隐藏功能), 优化了程序性能。

功能模块如下图:

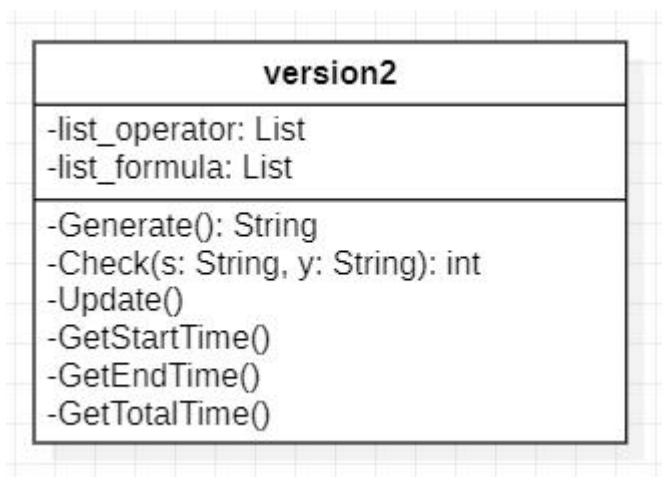


界面原型如下图:



(2)模块相关的设计类 (Design Class): 给出每个类的名字、属性、方法。

类中主要有 6 个方法, 执行生成算式、检查、换题、获取时间等功能。类图如下:



(3)所需的数据结构。

List: 用来存储已做的题, 以及 “+” ‘-’

String: 用来临时存储生成的式子

3.4 结对编程过程记录

(1) 角色切换与任务分工

表 2-1 第 2 轮迭代过程结对编程角色与任务分工

日期	时间(HH:MM - HH:MM)	驾驶员角色	领航员角色	本段时间的任务
10.20	18:00-19:30	尹大通	蔡建华	正确 (错误) 作答题数的记录、避免题目重复
10.20	19:50-20:10	蔡建华	尹大通	计时功能
10.20	20:15-21:00	蔡建华	尹大通	测试, 调整界面布局

【注意】该表格可自行增加更多的行。

(2) 工作日志

由领航员负责记录, 记录结对编程期间的遇到的问题、两人如何通过交流合作解决每个问题的。

表 2-2 第 2 轮迭代过程结对编程工作日志

日期/时间	问题描述	最终解决方法	交流过程
10.20/18:20 -18:30	避免题目重复应该如何实现	将已经练过的题目保存起来, 生成新的算式时再判断该算式是否出现过。若已出现过,	由于要避免题目重复, 因此我们认为就得知道之前已经出现过的题目。所以我们将出现过的算式保存在数

日期/时间	问题描述	最终解决方法	交流过程
		则重新出题。	组中, 生成新算式时判断该算式是否有在数组中出现。
10.20/19:20 -19:30	正确题数、错误题数的确定策略	点击“检查”时不记录正误, 点击“下一题”时再对输入的答案确定正误, 然后依此更新记录	点击“检查”时会对答案进行比对, 但不会切题。如果对同一个输入重复的答案反复地点击“检查”, 则正误题数记录会与实际做的题目数量不符。因此, 我们设定点击“下一题”时才更新正误题数的记录。
10.20/20:50 -21:00	UI 界面不够协调	进行多次的调试, 选择了布局较为平衡的界面	由于对界面的外观设计不是很熟悉, 因此我们进行多次的实际调试, 确定了较为协调的构件布局界面

【注意】该表格可自行增加更多的行

(3) 结对编程工作现场照片

请其他同学帮助拍摄结对编程现场照片至少 2 张。



第 2 次迭代结对编程现场照片 1



第 2 次迭代结对编程现场照片 2

3.5 单元测试

对本轮迭代所开发的每个类进行单元测试。

表 2-3 第 2 轮迭代过程的单元测试记录

日期/时间	类名: 方法名	输入数据	输出数据	结论
10.20/19:20 -19:30	version2:GetStartTime	无	19:21 (开始时间)	正常

日期/时间	类名: 方法名	输入数据	输出数据	结论
10.20/19:20 -19:30	version2:GetEndTime	无	19:22 (结束时间)	正常
10.20/19:20 -19:30	version2:GetTotalTime	无	1 (已经练习的时间)	正常
10.20/19:20 -19:30	version:check	“17 + 35”, ”52”	1	正常

3.6 集成测试

对本轮迭代所开发功能模块进行测试。

表 2-4 第 2 轮迭代过程的集成测试记录

日期/时间	功能名称	测试用例	测试结果描述	结论
10.20/21:00 -21:10	开始时间	在 21:01 点击 “开始”	21:01	正常
10.20/21:00 -21:10	结束时间	在 21:02 点击 “结束”	21:02	正常
10.20/21:00 -21:10	练习时间	在 21:02 时查看实时练习时长	显示已练习 47 秒	正常
10.20/21:00 -21:10	正确题数	总共答对 2 题	显示答对 2 题	正常
10.20/21:00 -21:10	错误题数	总共答错 1 题	显示答错 1 题	正常

4 第 3 轮迭代: 增量 2

4.1 迭代目标

简要介绍本轮迭代的增量任务目标, 所需完成的任务。

- (1) 任务 1: 增加 “错题集” 功能。
- (2) 任务 2: 增加 “音乐” 功能。
- (3) 任务 3: 美化界面, 优化布局。

4.2 需求分析

“错题集” 功能对于学生学习的重要性还是很高的, 通过回顾自己做错的题目可以总结不足、反思学习不到位的地方, 强化自己的学习能力, 因此 “错题集” 功能有实现的现实必要性与合理性。

用户为小学一年级学生, 应该更要考虑到软件对学生的友好性, “音乐” 功能可以增加使用的体验舒适度, 因此这一功能的实现有其必要性与合理性。

简洁美观的界面是影响用户使用的一个非常重要的因素，因此在这一轮迭代中我们调整了界面布局，优化界面观感。

4.3 设计方案

针对本轮迭代的开发内容给出设计方案：

(1)详细设计方案：模块功能结构设计，界面原型设计等。

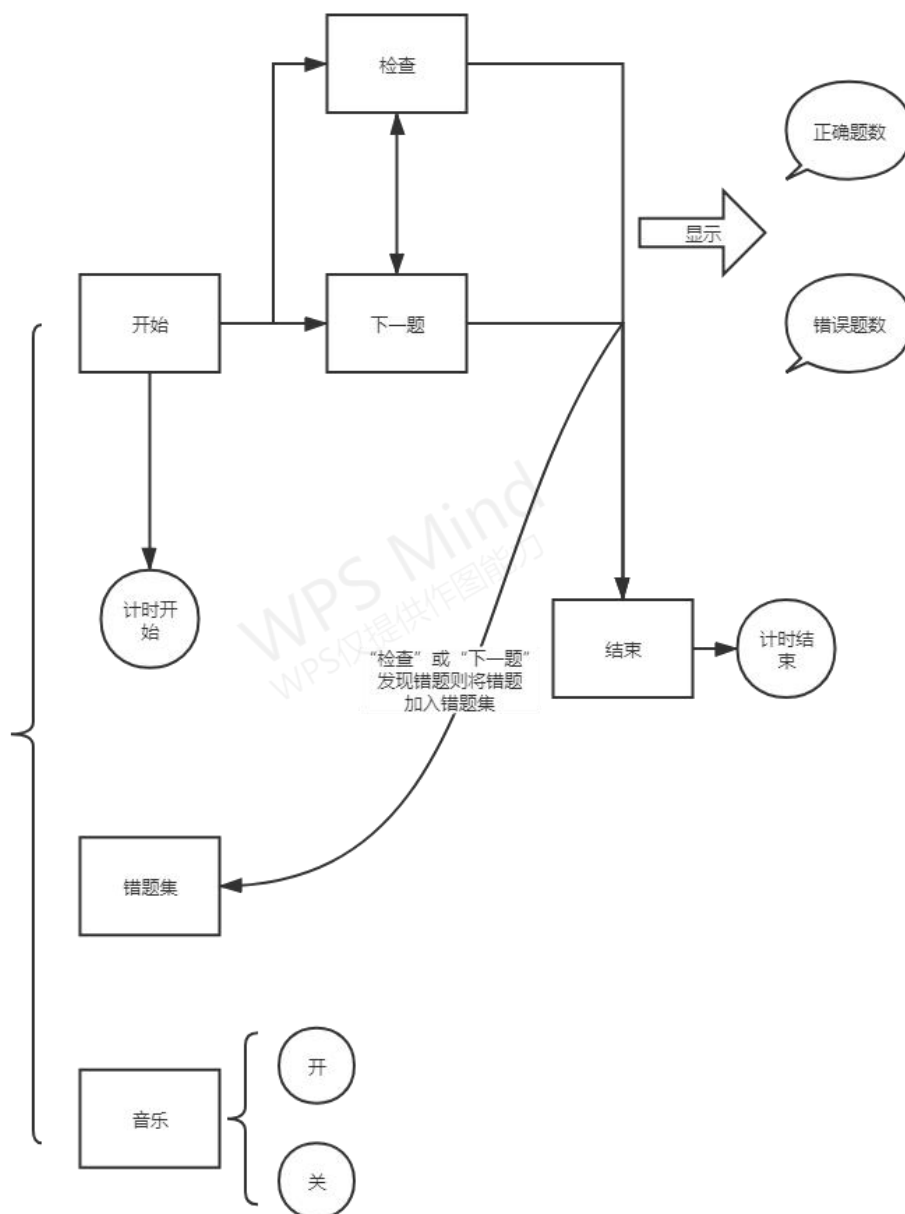
此轮迭代增加的功能有“错题集”与“音乐”，此外调整了布局、优化了界面。

“错题集”记录当次练习中的错题，在练习中或练习结束后可以查看错题集中的错题。

“音乐”功能可自主选择开或关。

另对界面进行了调整，使之更符合视觉观感。

功能模块如下图：



界面原型设计如下图:

100以内加减法练习

好好学习，天天向上

53 - 40 =

正答题数

0

错答题数

0

开始时间

13:12:49

结束时间

总时长

开始

结束

检查

下一题

请输入...

音乐

关

错题

0

错题集如下图:

错题集

15 + 14
24 + 39
81 - 53
73 + 0
94 - 50
64 - 33
75 - 7
64 - 50
64 - 39
59 + 53

(2)模块相关的设计类（Design Class）：给出每个类的名字、属性、方法。
类中主要有 8 个方法，执行生成算式、检查、换题、计时、音乐、错题等功能。类图如下：

version2
-list_operator: List -list_formula: List -list_wrong: List
-Generate(): String -Check(s: String, y: String): int -Update() -GetStartTime() -GetEndTime() -GetTotalTime() -CheckWrongQuestion() -GetStartTime()

(3)所需的数据结构。

List: 用来存储已做的题, 错题, 以及 “+” ‘-’

String: 用来临时存储生成的式子

Int: 用来做标志位, 标识歌曲是否播放

4.4 结对编程过程记录

(1) 角色切换与任务分工

表 3-1 第 3 轮迭代过程结对编程角色与任务分工

日期	时间(HH:MM - HH:MM)	驾驶员角色	领航员角色	本段时间的任务
10.21	18:00-18:30	尹大通	蔡建华	实现“错题集”功能
10.21	18:40-19:05	尹大通	蔡建华	实现“音乐”功能
10.21	19:30-20:00	蔡建华	尹大通	优化界面

(2) 工作日志

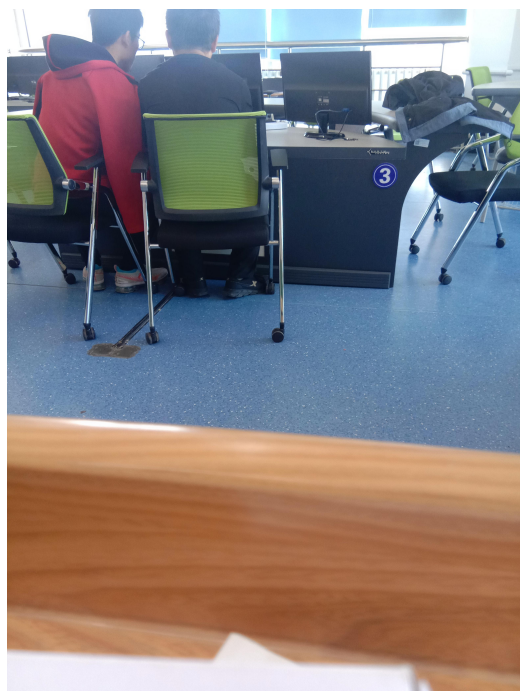
由领航员负责记录, 记录结对编程期间的遇到的问题、两人如何通过交流合作解决每个问题的。

表 3-2 第 3 轮迭代过程结对编程工作日志

日期/时间	问题描述	最终解决方法	交流过程
10.21/18:00 -18:05	错题集的显示	创建一个新窗口用于展示错题(当点击“错题集”按钮时展示)	由于错题集中的题目可能数量比较多, 如果显示在原窗口中, 界面可能会比较混乱, 因此我们选择用一个新的窗口来展示错题集, 以保证界面的协调。
10.21/18:40	播放音乐后无法进行	调用了 pygame 库中的	由于刚开始调用了 python

日期/时间	问题描述	最终解决方法	交流过程
-18:50	其他操作，只能等音乐播放完程序才能进行下一步操作	pygame.mixer.music 方法	中的 playsound 方法，出现了一旦播放音乐，程序将停留在 playsound 方法上直到音乐播放完，期间我们不能进行任何其他操作。通过查阅 python 文档，我们了解到还有 pygame 这个库，最后调用其中的 mixer.music 方法顺利解决了这个问题
10.21/19:40 -20:00	增加按钮后，界面整体不够协调	多次尝试后确定了比较满意的布局	由于一开始并没有很确定的界面布局设计，因此我们决定先构造一个界面原型，再在此基础上继续改进。经过多次的调整，确定了较为满意的界面布局

(3) 结对编程工作现场照片



第 3 次迭代结对编程现场照片 1



第 3 次迭代结对编程现场照片 2

4.5 单元测试

对本轮迭代所开发的每个类进行单元测试。

表 3-3 第 3 轮迭代过程的单元测试记录

日期/时间	类名：方法名	输入数据	输出数据	结论
10.21/18:30 -18:40	version3:checkWrong Question	“96 - 67”	“96 - 67”	正常

日期/时间	类名: 方法名	输入数据	输出数据	结论
10.21/18:30 -18:40	version3:checkWrong Question	"10 - 7", "89 + 48"	"10 - 7", "89 + 48"	正常
10.21/19:05 -19:15	version3:check	"63 + 48", "111"	1	正常

4.6 集成测试

对本轮迭代所开发功能模块进行测试。

表 3-4 第 3 轮迭代过程的集成测试记录

日期/时间	功能名称	测试用例	测试结果描述	结论
10.21/20:00 -20:20	音乐	无	点一下开始播放, 再点一下停止播放	正常
10.21/20:00 -20:20	错题集	练习过程中产生错题: "10 - 7", "89 + 48"	显示错题: "10 - 7", "89 + 48"	正常
10.21/20:00 -20:20	开始	点击“开始”	产生算式, 开始练 习, 并开始计时	正常
10.21/20:00 -20:20	结束	点击“结束”	结束练习, 并停止 计时	正常

5 项目总结

5.1 增量模型应用总结

- 增量划分是根据工作量划分还是根据用户需求来划分?

主要是根据用户需求来划分的。一个软件中, 用户的需求一般有很多, 而这些需求之中又有必需和非急需功能需求之分, 在时间急迫的增量开发过程中, 应该优先实现那些必需的基础功能需求。因此, 增量的划分不是根据工作量划分而是根据需求划分。当然, 划分增量过程中也要注意避免增量间工作量差距太大的情况。

- 第 1 个增量你们团队是如何确定的?

确定项目需求中所必须实现的基础功能。如我们实现的是 100 以内加减法, 因此我们认为首先应该实现: 随机生成练习式、检查答案、换题等功能, 这是软件等基本运行并满足最基本用户使用需求所应该实现的基础功能。于是, 第一个增量就这么确定了。

- 你认为增量模型的好处在哪里？有什么不好之处吗？它的适用场合是什么？

好处在于在时间紧迫的种种情况下，可以快速开发出一个能提供基础功能、满足最基本使用需求的初代版本软件。在抢占市场、应急情况下具有重要的作用。

不好之处：由于各个构件是逐步并入现有软件体系结构中的，所以加入构件必须不能破坏已有的构件，这就要求软件具备开放式的体系结构；在开发过程中，需求的变化时不可避免的，增量模型适应这种变化的能力又很强，但容易退化为边做边改模型。

适合场合：技术风险较大、用户需求较为稳定的软件系统。

5.2 结对编程总结

对结对编程过程进行总结，包括但不限于以下几方面的内容：

- 与 2 人分别编程相比，是否体验到编程效率的提高、编程质量的提高？

体验到了编程效率与质量的提高。两个人一起编程更容易发现潜在的 bug，同时能够集思广益，选择更合适的编程方法。

同时编程时对于代码的编写思路更清晰，效率更高。

- 你认为，你们 2 人本次合作是否真的提高了效率和质量？有什么成功的体验和需要反省的不足或教训？

提高了效率和质量。结对编程在写代码时效率更高，质量更好，但是两人在编程前要合理分配好任务，同时要事先对于编程方案达成共识。

- 你认为结对编程的优势在哪里、有什么不适应之处？它的适用场合是什么？

结对编程优势在于质量与效率。结对编程需要两个人之间有一定的沟通交流，需要两人事先达成一致的编程方案。适用于对项目质量、安全性要求高的项目，结对编程一定程度上降低了代码上出现较大错误的可能性。