

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

2021 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
沈城有	1190200526	1190200526@stu.hit.edu.cn
傅浩东	1190202105	1190202105@stu.hit.edu.cn

## 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.....	3
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.....	4
2.3 设计方案.....	4
2.4 结对编程过程记录.....	6
2.5 单元测试.....	7
2.6 集成测试.....	7
3 第 2 轮迭代：增量 1.....	8
3.1 迭代目标.....	8
3.2 需求分析.....	8
3.3 设计方案.....	8
3.4 结对编程过程记录.....	9
3.5 单元测试.....	10
3.6 集成测试.....	11
4 第 3 轮迭代：增量 2.....	11
4.1 迭代目标.....	11
4.2 需求分析.....	11
4.3 设计方案.....	12
4.4 结对编程过程记录.....	13
4.5 单元测试.....	15
4.6 集成测试.....	15
5 项目总结.....	16
5.1 增量模型应用总结.....	16
5.2 结对编程总结.....	16

## 1 项目概述

(1) 项目名称: 仿真计算器 (常规计算器)

(2) 项目简要介绍:

仿真计算器项目是一个使用 Java 的计算器实现, 其主要功能包含加、减、乘、除、阶乘、开方等基本运算 (其中要包括报错、括号处理等功能), 进阶运算包括三角函数、指数类函数等计算, 拓展功能包含记忆、历史记录、清除、声音交互等功能。

### 1.1 总体需求分析

(1) 项目要开发的软件的目的和意义:

个人计算机通常带有一个计算器实用程序, 用于模拟计算器的外观和功能, 使用图形用户界面来描绘计算器, 也就是我们所做的仿真计算器, 主要目的是为了给使用者提供一个能够用于各式各样的系统上的计算 (例如 Windows、Android 等), 用于解决在工作学习中需要进行的简易数值计算的需求, 同时减少物理固件带来的不便, 可以满足临时需要计算的需求。

(2) 软件系统的涉众分析:

考虑到仅仅设计了一个简单的仿真计算器, 我们关心的主要问题是基础的计算功能, 所以该系统主要与以下人员相关: 账务, 学生等; 其关心的核心内容就在于能够进行基础的几种计算, 尤其是对于财务更加关心的是计算的准确性, 正确的计算结果关系到他们的工作结果与公司财务收入等; 对于学生来说关心的是计算功能的完整性, 从基础计算一直到科学计算, 对学生这个群体来说需求是日益增大的。

(3) 软件系统的功能需求:

用户使用计算器首先就需要仿真计算器软件能够执行基本的加减乘除运算功能, 此时除了键盘输入, 更加期待的是能够使用按键来进行输入, 当然希望如果输入错误会有提示, 如果超出计算能力范围也会进行提示。

当用户遇到了更加复杂计算, 其中涉及到三角函数、指数类函数、幂函数运算, 用户需要将这些函数加入到计算算式中。

在这之后, 如果用户需要输入一个较长的公式来一次性得出结果, 软件需要能够对复杂计算式进行计算。

计算结束后, 用户可能需要重新查看计算的结果, 所以需要软件具备历史查询功能。

(4) 软件系统的非功能需求:

**性能要求:** 对于基本的简单运算做到实时响应, 对于三角函数等科学计算计算做

到 0.1s 内响应，对于能力内较复杂幂函数计算和阶乘计算做到 0.5s 内响应。

**界面要求：**软件使用仿真计算机界面，包含基本按钮和菜单键，用户输入计算式时，软件界面能够同步显示。使用高对比度色调，确保用户能够清楚看到计算过程和结果。在按动键盘时，计算器界面响应的按钮也会做出反应。

**技术指标：**

- 支持计算式长度：

视具体情况而定，至少支持 10 字符以上计算式计算。

- 支持计算种类：

基本运算（加减乘除），平方运算，开方运算，常用函数，幂函数，阶乘，指数运算，模运算等。

- 可靠性需求：

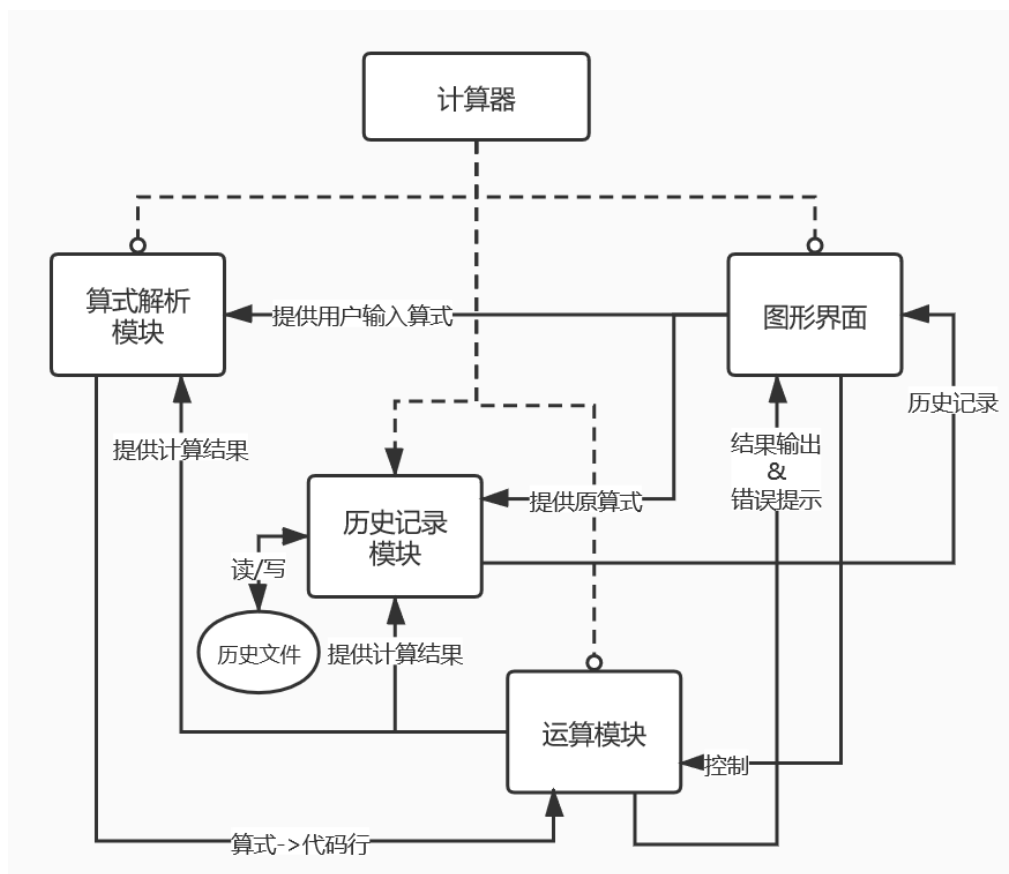
具有检测异常输入的能力，对于非计算式以外的符号输入能够做出识别。对于内容不正确计算式（比如分母除 0）能够识别并提示用户，对于括号不匹配或格式不正确计算式不予计算。

- 响应速度：

一般计算做到实时响应，复杂计算在 0.5s 内响应。

## 1.2 软件系统总体设计

(1) 软件系统功能结构设计方案：



(2) 软件系统相关的数据结构设计:

- PyQt5 图形库: Python 的图形库之一, 在系统中实现用户界面;
- math 库: 提供 Python 的数学相关支持, 在系统中执行计算功能(核心), 并用于非法算式的错误提示;
- pyttsx3 库: 语音引擎, 利用系统内置语音实现文字转语音输出;
- 文件 IO: 使用 Python 所支持的文件读写操作实现历史记录的添加、读取和展示以及关于信息的读取和展示;
- string: Python 字符串类型, 在系统中用于算式表示、处理和输出。

### 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:

**任务列表:**

- a) 实现基本的用户 GUI 界面;
- b) 实现算式检查、解析和错误提示功能;
- c) 实现基本的计算功能(包括四则运算、开方等)及退格、清零功能;
- d) 调试以上功能, 测试其正确性和鲁棒性;
- e) 将源程序打包为可执行的 exe 文件。

**进度安排:**

2021.10.15 上午(9:00 - 11:00)完成任务 a)、b), 其余时间完成任务 c)、d)、e), 此轮迭代计划完成项目的基础开发。

(2) 第 2 轮迭代的任务:

**任务列表:**

- a) 调整用户界面, 提供新的操作、控制支持(如括号插入);
- b) 实现较复杂的计算功能(如三角函数、对数等);
- c) 调试以上功能, 测试其正确性和鲁棒性;
- d) 将源程序打包为可执行的 exe 文件。

**进度安排:**

2021.10.16 上午(9:00 - 11:00)完成任务 a)、b), 下午(14:00 - 17:00)完成任务 c)、d)。

(3) 第 3 轮迭代的任务:

**任务列表:**

- a) 实现计算历史的记录和查询功能;
- b) 实现计算历史以可读形式写入文本文件以长期存储的功能;
- c) 实现语音朗读计算器当前显示内容的功能;
- d) 在用户界面菜单栏中添加以上功能的入口, 调试以上功能, 测试其正确

性和鲁棒性;

e) 将源程序打包为可执行的 exe 文件。

### 进度安排:

2021.10.17 上午 (9:30 - 11:00) 完成任务 a)、b), 下午 (14:00 - 17:00) 完成任务 c)、d), 晚上 (18:30 - 21:00) 调整、修改程序并完成任务 e)。

## 1.4 开发环境与工具

开发语言: Python 3.9

开发环境: Windows 10 / Windows 11

开发工具: Visual Studio Code 1.61.1 Qt Designer

打包工具: Pyinstaller

## 2 第 1 轮迭代: 基本系统

### 2.1 迭代目标

- (1) 任务 1: 实现基本的用户 GUI 界面;
- (2) 任务 2: 实现算式检查、解析和错误提示功能;
- (3) 任务 3: 实现基本的计算功能及退格、清零功能;
- (4) 任务 4: 调试以上功能, 测试其正确性和鲁棒性;
- (5) 任务 5: 将源程序打包为可执行的 exe 文件。

### 2.2 需求分析

本轮迭代所开发模块的详细需求分析如下:

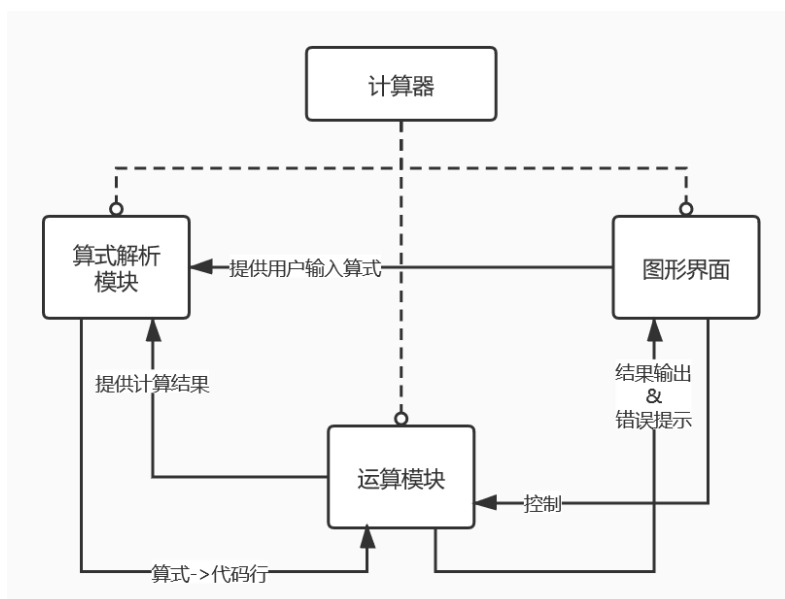
- 用户希望通过图形界面而不是命令行交互的形式使用计算器, 故需要实现基本的 GUI 界面;
- 用户可能输入存在错误的算式, 这会导致程序无法执行计算, 故应检查用户输入的算式, 在发现错误时允许用户重新输入;
- 本项目实现的是计算器, 保证基本的计算功能和一些辅助功能是项目的核心和重点。

### 2.3 设计方案

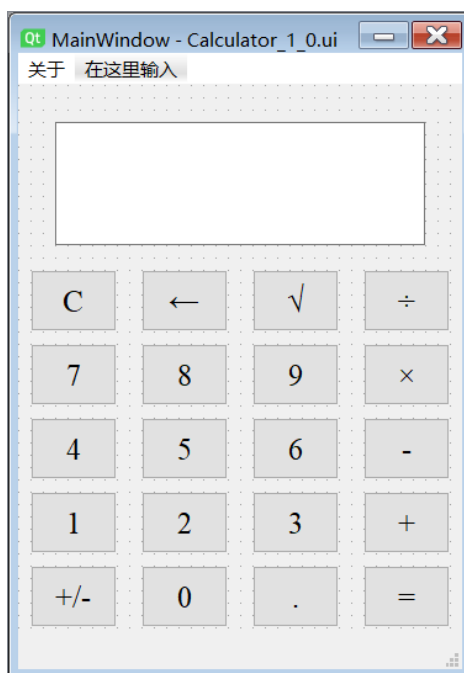
(1) 详细设计方案:

#### 模块功能结构设计:

本次迭代完成的系统主要由算式解析模块、运算模块和图形界面三个部分组成, 模块间关系见下页图:



界面原型设计:



(2) 模块相关的设计类 (Design Class):

● Calculator\_1\_0 (主类):

方法	说明
setupUi()	用于初始化系统主界面, 包括窗口、按键、菜单栏、状态栏等。
retranslateUi()	初始化图形化界面中的所有文本。
press_btn()	在按键被用户按下时触发, 执行系统功能。

● Calculator\_1\_0\_launcher (启动器):

无具体方法, 用于在软件启动时进行初始化, 并负责在软件关闭后的进程终止工作。

(3) 所需的数据结构:

- a) PyQt5 图形库: Python 的图形库之一, 在系统中实现用户界面;
- b) math 库: 提供 Python 的数学相关支持, 在系统中执行计算功能(核心), 并用于非法算式的错误提示;
- c) string: Python 字符串类型, 在系统中用于算式表示、处理和输出。

## 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
2021.10.15	8:30 - 11:00	傅浩东	沈城有	实现基本的用户 GUI 界面。
2021.10.15	14:00 - 17:00	沈城有	傅浩东	实现算式检查、解析和错误提示功能和基本计算功能。
2021.10.15	18:00 - 20:00	傅浩东	沈城有	实现退格、清零功能并对以上所有功能进行测试。
2021.10.15	20:30 - 22:00	沈城有	傅浩东	将源程序打包为可执行的 exe 文件, 总结此轮迭代成果, 整理报告相关内容。

### (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
2021.10.15 8:50	Qt Designer 环境配置报错。	调整扩展中的路径设置, 重新定位工具模块位置。	傅浩东首先发现此问题, 由沈城有进行排查和在网上搜索相关问题的解决办法, 最终合作解决。
2021.10.15 14:30	算式处理存在缺陷, 存在数组越界访问问题。	重新设计算式处理相关逻辑实现, 增加越界检测。	沈城有首先发现此问题, 由傅浩东提出了相关的解决策略, 得以最终解决。
2021.10.15 19:00	测试中发现用户界面输出在一些操作后会混乱。	重新设计输出部分逻辑实现, 补充在一些操作后恢复输出内容的操作。	傅浩东在测试程序运行效果时发现此问题, 由沈城有重新检查了其编写的相关代码, 确定了问题位置, 并最终改正。

### (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
2021.10.15 16:00	Calculator_1_0	NULL	正确显示界面	True
2021.10.15 16:05	Calculator_1_0_launcher	NULL	按钮与功能正确绑定, 运算正常	True
2021.10.15 16:05	setupUi (方法)	NULL	显示原始界面	True
2021.10.15 16:10	retranslateUi (方法)	NULL	初始文本显示	True
2021.10.15 16:10	press_btn (方法)	NULL	按键正常且能够与绑定功能正确交互	True

## 2.6 集成测试

对本轮迭代所开发功能模块进行测试。

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
2021.10.15 20:00	加	123+45+11+77	334	True
2021.10.15 20:00	减	111-78-34	-1	True
2021.10.15 20:00	乘	7×8	56	True
2021.10.15 20:00	除	89÷45	1.9777777777777779	True
2021.10.15 20:05	根号	√9	3.0	True
2021.10.15 20:05	回退	789×8+5√←	789×8+5	True
2021.10.15 20:10	清除	789×8+5	0	True

## 3 第 2 轮迭代：增量 1

### 3.1 迭代目标

- (1) 任务 1: 调整用户界面, 提供新的操作、控制支持;
- (2) 任务 2: 实现较复杂的计算功能;
- (3) 任务 3: 调试以上功能, 测试其正确性和鲁棒性;
- (4) 任务 4: 将源程序打包为可执行的 exe 文件。

### 3.2 需求分析

本轮迭代所开发模块的详细需求分析如下:

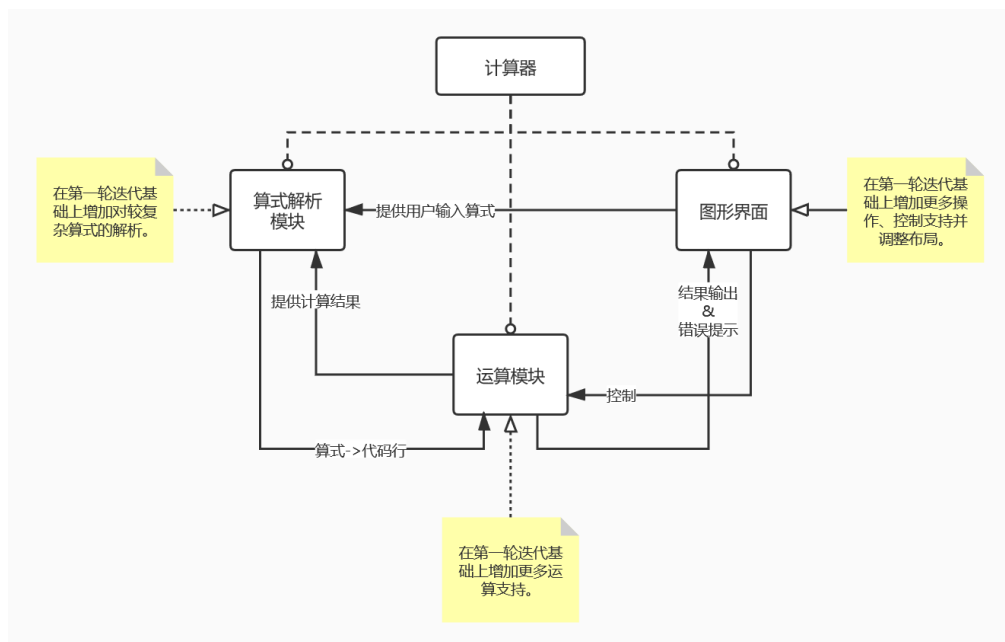
- 用户希望计算器能提供更复杂的计算功能, 满足不同场景的使用, 故需要通过进一步完善、增强运算模块和算式解析模块来实现;
- 为使用户能使用新增的功能, 我们需要对界面进行调整和完善, 添加更多的按钮并保证简洁和美观。

### 3.3 设计方案

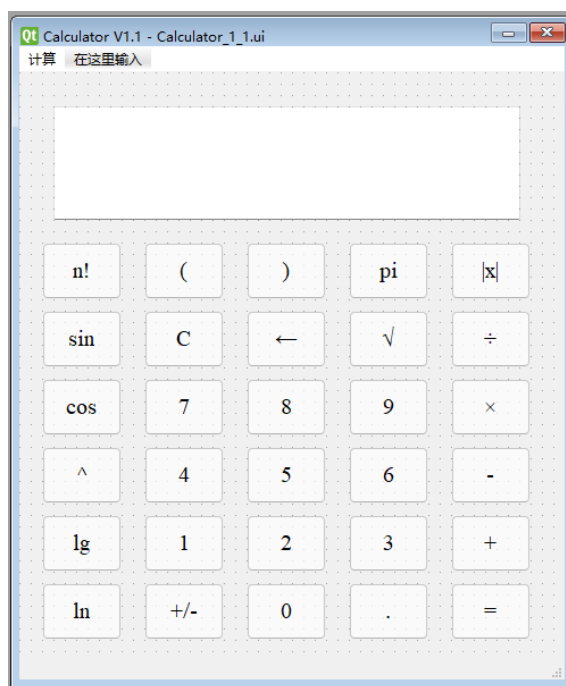
针对本轮迭代的开发内容给出设计方案:

- (1) 详细设计方案:

模块功能结构设计:



界面原型设计:



## (2) 模块相关的设计类 (Design Class):

### ● Calculator\_1\_1 (主类):

方法	说明
setupUi()	用于初始化系统主界面, 包括窗口、按键、菜单栏、状态栏等。
retranslateUi()	初始化图形化界面中的所有文本。
press_btn()	在按键被用户按下时触发, 执行系统功能。

### ● Calculator\_1\_1\_launcher (启动器):

无具体方法, 用于在软件启动时进行初始化, 并负责在软件关闭后的进程终止工作。

## (3) 所需的数据结构:

- PyQt5 图形库: Python 的图形库之一, 在系统中实现用户界面;
- math 库: 提供 Python 的数学相关支持, 在系统中执行计算功能 (核心), 并用于非法算式的错误提示;
- string: Python 字符串类型, 在系统中用于算式表示、处理和输出。

## 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
2021.10.16	9:00 - 11:00	傅浩东	沈城有	调整用户界面, 提供新的操作、控制支持。
2021.10.16	14:00 - 15:20	沈城有	傅浩东	实现较复杂的计算功能。

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
2021.10.16	15:30 - 17:00	傅浩东	沈城有	测试、调整程序, 打包, 总结成果。

## (2) 工作日志

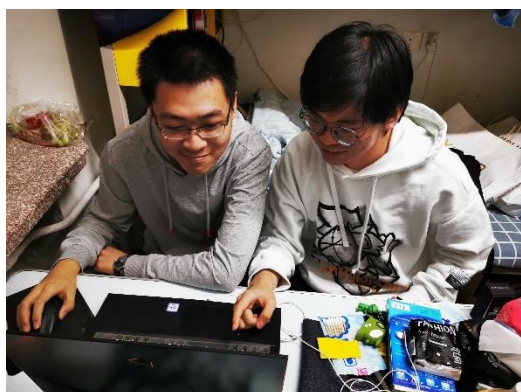
由领航员负责记录, 记录结对编程期间的遇到的问题、两人如何通过交流合作解决每个问题的。

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
2021.10.16 10:00	菜单项触发事件无响应。	参考网上相关技术博客和教程进行操作。	傅浩东在作为驾驶员进行界面设计、编程时发现此问题, 通过一同参考网络博客和教程得以解决。
2021.10.16 15:50	程序在执行含三角函数的相关计算时无响应。	根据沈城有提出的建议由傅浩东编程解决。	傅浩东发现了这一问题, 通过共同检查相关代码发现潜在的死循环, 避免死循环后解决。

## (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
2021.10.16 16:00	测试上一次增量中的类与方法	NULL	功能实现完整正确	True
2021.10.16 16:05	setUpUi (方法)	NULL	显示原始界面且增加的功能显示齐全	True
2021.10.16 16:10	retranslateUi (方法)	NULL	初始文本显示, 新增按钮文本显示正确	True
2021.10.16 16:10	press_btn (方法)	NULL	按键正常且能够与绑	True

日期/时间	类名: 方法名	输入数据	输出数据	结论
			定功能正确交互, 新增功能正确 (见如下功能测试)	

### 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
2021.10.16 16:30	上一增量的功能	上述相同	结果相同	True
2021.10.16 16:40	Cos 函数	cos(4)	-0.6536436208636119	True
2021.10.16 16:40	Cos 函数	cos(2×pi)	1.0	True
2021.10.16 16:40	Sin 函数	sin(4)	-0.7568024953079282	True
2021.10.16 16:40	Sin 函数	sin(2.5×pi)	1.0	True
2021.10.16 16:45	^次数	7^4	2401	True
2021.10.16 16:45	1/Lg (10 底)	1/lg(138)	0.46731612377303067	True
2021.10.16 16:45	Ln 对数 (e 底)	ln(7)	1.9459101490553132	True
2021.10.16 16:45	n!阶乘	(5)!	120	True
2021.10.16 16:45	Pi 数Π	2×pi	6.283185307179586	True
2021.10.16 16:50	x 绝对值	abs(-7-6)	13	True
2021.10.16 16:50	括号	(7-9)×13	-26	True

## 4 第 3 轮迭代: 增量 2

### 4.1 迭代目标

- (1) 任务 1: 实现计算历史的记录和查询功能;
- (2) 任务 2: 实现计算历史以可读形式写入文本文件以长期存储的功能;
- (3) 任务 3: 实现语音朗读计算器当前显示内容的功能;
- (4) 任务 4: 在用户界面菜单栏中添加以上功能的入口, 调试以上功能, 测试其正确性和鲁棒性;
- (5) 任务 5: 将源程序打包为可执行的 exe 文件。

### 4.2 需求分析

本轮迭代所开发模块的详细需求分析如下:

- 用户希望计算器能够支持对历史计算的记录和查看, 有时还需要导出为文本文件备用, 故我们应设计实现历史相关功能;
- 用户希望计算器能够朗读算式及结果, 提高一些场景下的效率, 故我们

应设计实现语音朗读；

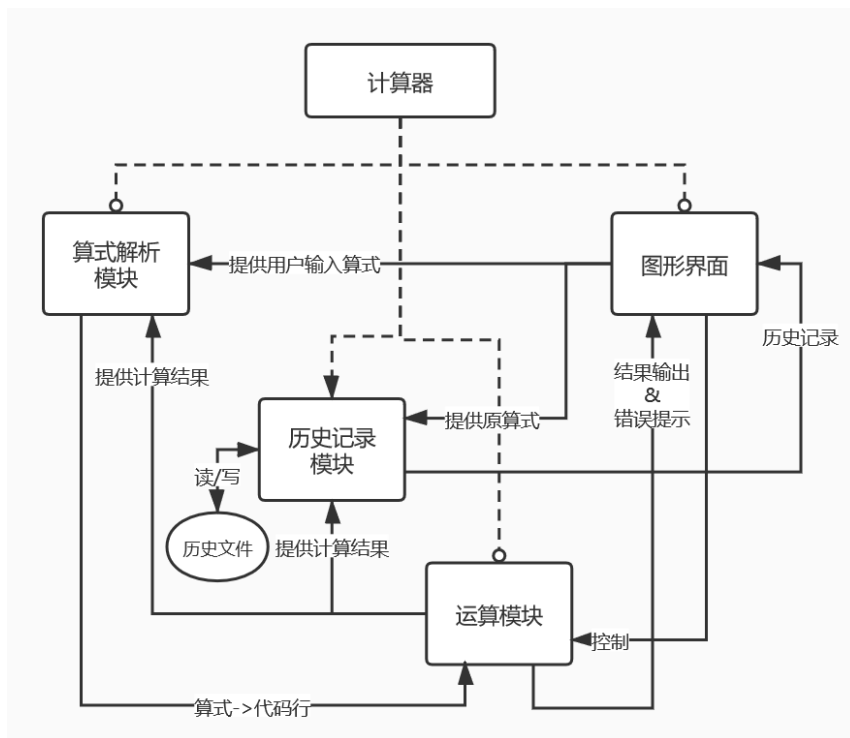
- 为便于用户使用以上功能，我们计划将以上功能的入口放入菜单栏，并设置对应快捷键。

## 4.3 设计方案

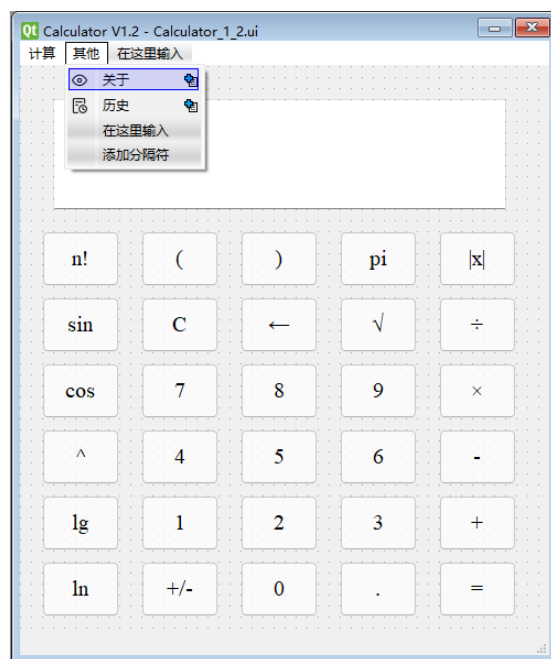
针对本轮迭代的开发内容给出设计方案：

(1) 详细设计方案：

模块功能结构设计：



界面原型设计：



## (2) 模块相关的设计类 (Design Class):

## ● Calculator\_1\_2 (主类):

方法	说明
setupUi()	用于初始化系统主界面（包括窗口、按键、菜单栏、状态栏等）及后续朗读功能的语音引擎。
retranslateUi()	初始化图形化界面中的所有文本。
press_btn()	在按键被用户按下时触发，执行系统功能。

## ● Calculator\_1\_2\_launcher (启动器):

无具体方法，用于在软件启动时进行初始化（本次迭代添加了对菜单项的事件绑定语句、关于子窗口实例化语句和根据历史文件内容初始化历史窗口的函数），并负责在软件关闭后的进程终止工作。

## ● Sound (语音引擎类):

方法	说明
__init__()	实例化对象时调用。
sound_init()	初始化语音引擎。
make_voice()	执行语音朗读指定文本内容。

## ● Ui\_about (关于子窗口类):

方法	说明
setupUi()	用于初始化关于子窗口。
retranslateUi()	初始化关于子窗口中的文本。

## ● Ui\_history (历史子窗口类):

方法	说明
setupUi()	用于初始化历史子窗口。
retranslateUi()	读取历史文件内容用于初始化历史子窗口中的文本。

## (3) 所需的数据结构:

- PyQt5 图形库: Python 的图形库之一，在系统中实现用户界面；
- math 库: 提供 Python 的数学相关支持，在系统中执行计算功能（核心），并用于非法算式的错误提示；
- pyttsx3 库: 语音引擎，利用系统内置语音实现文字转语音输出；
- 文件 IO: 使用 Python 所支持的文件读写操作实现历史记录的添加、读取和展示以及关于信息的读取和展示；
- string: Python 字符串类型，在系统中用于算式表示、处理和输出。

## 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
2021.10.17	9:30 - 11:00	傅浩东	沈城有	实现计算历史的记录和查询以及以可读形式写入文本文件的功能。
2021.10.17	14:00 - 17:00	沈城有	傅浩东	实现语音朗读计算器当前显示内容的功能，并为各功能在界面添加入口。
2021.10.17	18:30 - 21:00	傅浩东	沈城有	测试功能正确性、鲁棒性，根据测试结果调整程序，打包，总结成果，撰写报告。

### (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
2021.10.17 10:10	历史文件路径不存在问题。	通过一同尝试、讨论和总结找到了问题所在，最终独立解决。	傅浩东在作为驾驶员进行历史功能相关实现时遇到此问题，通过与沈城有共同讨论、测试最终解决。
2021.10.17 16:30	语音引擎初始化失败问题。	傅浩东检查相关代码，找到问题，提供相关思路由沈城有编程解决。	沈城有发现了这一问题，傅浩东检查了相关代码，提出了具体的解决思路。
2021.10.17 19:00	历史文件读取时无法找到本次软件运行的记录。	两人共同回顾相关代码，通过分析、调整方法执行顺序和调用关系解决。	傅浩东通过测试发现这一问题并同沈城有重新分析了相关代码，最终由沈城有提出了问题位置和解决思路，由傅浩东修改代码实现。

### (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 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
2021.10.17 20:00	上一次增量中的类与方法	NULL	功能实现完整正确	True
2021.10.17 20:30	Ui_about	NULL	显示关于界面	True
2021.10.17 20:30	Ui_history	NULL	显示历史界面	True
2021.10.17 20:40	Sound	NULL	根据文字播放语音	True
2021.10.17 20:40	make_voice	NULL	将字符替换为文字, 并且发出语音	True
2021.10.17 20:40	sound_init	NULL	初始化语音包	True

## 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
2021.10.17 20:40	上一增量的功能	上述相同	结果相同	True
2021.10.17 20:45	点击显示关于界面	点击菜单	正确显示关于界面	True
2021.10.17 20:45	快捷键关于界面	快捷键	正确显示关于界面	True
2021.10.17 20:50	点击显示历史界面	点击菜单	正确显示历史界面且第一行为最近一次计算结果	True
2021.10.17 20:50	快捷键历史界面	1+2=3 快捷键	正确显示且第一行为最近一次计算结果	True
2021.10.17 20:50	点击播放结果	1+2=3 点击菜单	正确播放	True
2021.10.17 20:50	快捷键播放结果	1+2=3 快捷键	正确播放	True

## 5 项目总结

### 5.1 增量模型应用总结

对增量模型应用过程进行总结，包括但不限于以下几方面的内容：

- 增量划分是根据工作量划分还是根据用户需求来划分？

答：两个方面都应考虑。用户需求决定了软件开发的主要任务，但由于用户需求的内容较模糊、宽泛，距离具体实现有一定差距，在实际开发实现前需要对其进行分析和处理。这一过程就包含衡量用户需求的可行性、优先级、风险和对应的工作量。工作量太大难以在一次迭代中完成，过小又会影响开发效率，故二者需要兼顾。

- 第 1 个增量你们团队是如何确定的？

答：我们参考了 Windows 系统自带的常规计算器，对其上进行了功能的测试和简化，初步建立了原型并确定了我们要首先开发的部分，即简单四则运算、开方和部分输入控制。

- 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？

答：

**好处：**

- a) 可以将一个大的任务划分为多个小任务并逐次实现，降低开发难度；
- b) 能够很好地适应需求的增加；
- c) 每次迭代的条理比较清晰，目标比较明确。

**缺点：**

- a) 由于各个构件是逐渐并入已有的软件体系结构中的，所以加入构件必须不破坏已构造好的系统部分，这对软件系统的模块化程度有一定要求，如果已经实现的需求发生变化难以适应；
- b) 增量模型的灵活性可以使其适应需求增加的能力大大优于瀑布模型，但也可能退化为边做边改的模式，从而使软件过程的控制失去整体性。

**适用场合：**用户急需使用软件的核心功能，后续延伸功能暂时不急需。总的来说，适合于快速迭代、开发较简单且风险较低的小项目。

### 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？

答：双方轮流担任领航员和驾驶员，可以有效提高编程质量，也能更快、更容易发现错误和潜在的问题，但是对于这个小项目而言编程的效率并没有

明显的提升。

● 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？

答：本次 2 人合作并没有真的提高效率，但是提高了质量。**体验：**结对编程可以很好地提高代码的准确率，两人共同工作会稍微降低疲惫感。**不足与教训：**代码模块化不好，结对编程需要在思维、经验和能力上紧密配合。

● 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？

答：**优势：**结对编程可以很好地双方互相学习；降低错误量；两人相互讨论、结合两人知识可以快速解决问题及进行知识共享；领航员对项目进程的影响不大，交换编程不会增加疲惫感；省下了许多修改与测试的时间。**不适应：**写代码速度降低；两人想法不一样产生矛盾，项目难以禁止；沟通过程中，有可能会引起争执；一个人的工作两个人做，浪费人力资源。**适应场合：**快速设计、实现软件的场合并且需要避免错误的时候，降低代码错误；结对编程的初始质量会较高，所以也适用于初步快速设计软件。