

软件设计文档

1.引言

1.1 编写目的

本系统介绍了在 visual studio 2019 环境下采用“自上而下地总体规划，自下而上地应用开发”的策略开发一个自动考试系统的开发过程。

1.2 背景

在社会对考核日渐注重的今天，考试被认为是一种相对来说科学合理的评价方式，因此开发一个方便好用的、可以随时出题答卷的系统至关重要。

1.3 定义

自动考试系统：出题人员可以通过该系统实现自主出题，出好的试卷可以保存至硬盘。答卷人员可以打开硬盘中已有的试卷进行答题，提交试卷之后系统自动核算分数并反馈给答题者。

1.4 参考文献

CSDN 博客上的相关文章以及 Google 上的相关文章。

2.总体设计

2.1 需求规定

(1) 出题人员可以在出题界面中出题，首先选择题型，系统针对选择的题型弹出有针对性的对话框，之后出题人员填入相应的题目描述和分值。

(2) 出题人员在出题的过程中可以通过题目列表来浏览已出题目的题号、题型、分值。

(3) 系统自动核算试卷总分，若不够 100 分则继续弹出下一题的出题界面，若超过 100 分则提醒用户修改分值，若满 100 分则自动提交。

(4) 出题结束后出题人员可以双击出题列表浏览已出试题，确认试题无误后可以保存至硬盘。

(5) 答题人员可以从电脑中打开已有的试卷进行答题。

(6) 答题时可以通过“上一题”和“下一题”按钮来浏览试题，答题结束后点击“交卷”按钮提交。

(7) 提交之后系统按照提前设定的评分准则判卷并核算总分，之后将总分反馈给新生。

2.2 运行环境

visual studio 2019

2.3 基本设计概要和处理流程

设计概要：通过 MFC 单文档设计，实现数据的交互和处理分离，数据储存在 Doc 文

件中，交互的控制主要在 View 中完成，与用户进行数据的传递主要通过对对话框完成。对话框是 View 类中的一个对象，获得的信息通过对对话框的数据成员传到 View 中，View 中的数据再通过 GetDocument 函数所获取的指针传入 Doc 文件中。

处理流程：

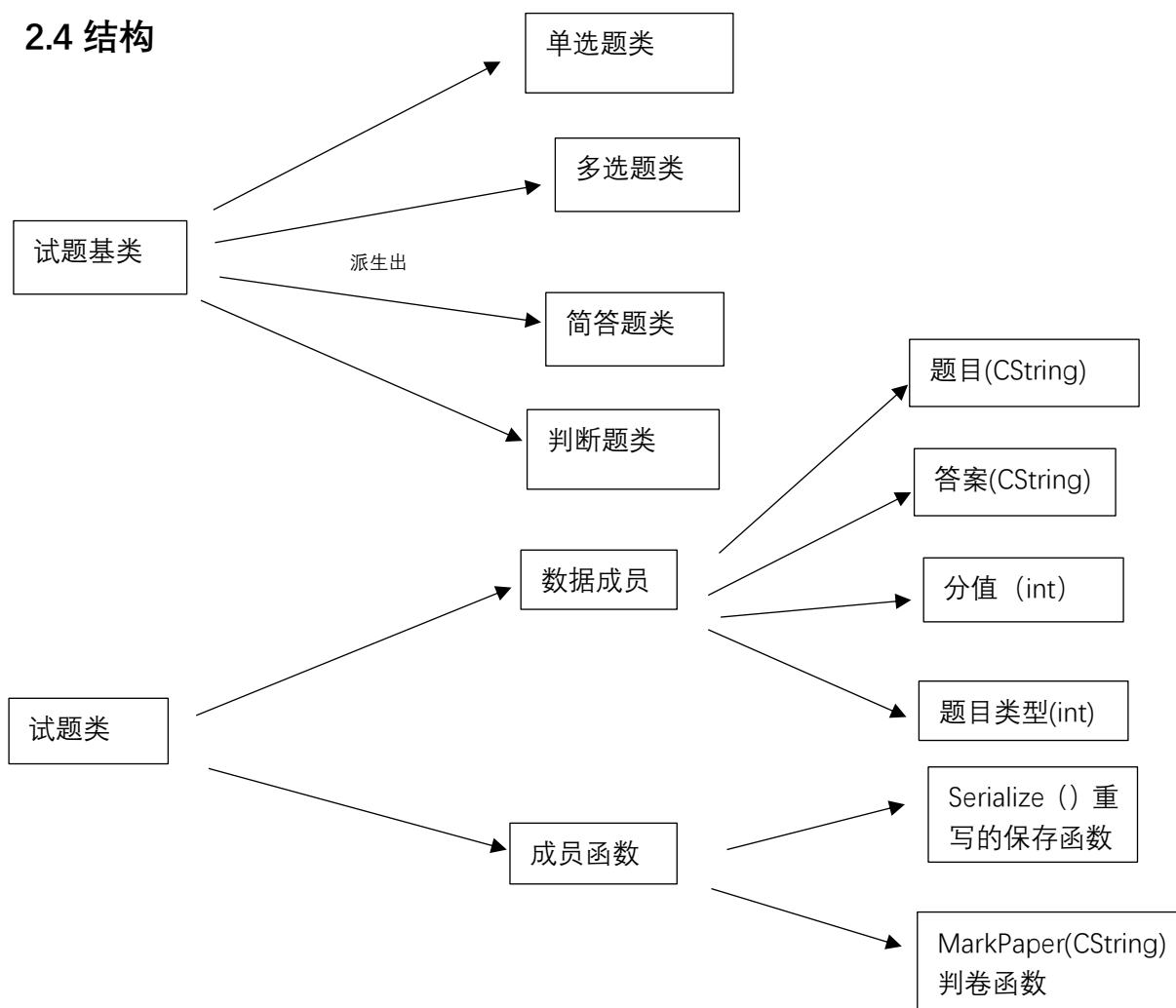
- 出题：
- 1) 出题人员点击出题按钮，触发 View 类里的响应函数
 - 2) 响应函数切换视图为列表视图，创建出题对话框的对象，并开始弹出
 - 3) 出题过程本质上是一个循环，根据用户选择的题型弹出对应对话框，再将对话框中的数据不断的传到 View 函数中，再通过文档的指针传入文档中。
 - 4) 当 Doc 内部函数核算分数达到 100 分，通知 View 类结束循环，出题完毕。
 - 5) 用户双击列表视图可以触发列表视图中的响应函数，创建查看试题的对话框，并将 Doc 中的数据放入对话框中。
 - 6) 确认试题无误后点击保存触发 Doc 的 Serialize 函数，将数据存入文件的同时切换回原有视图。

答题：1) 答题人员点击打开按钮打开试卷，触发 Serialize 函数，将文件中的数据重新放入 Doc 中。

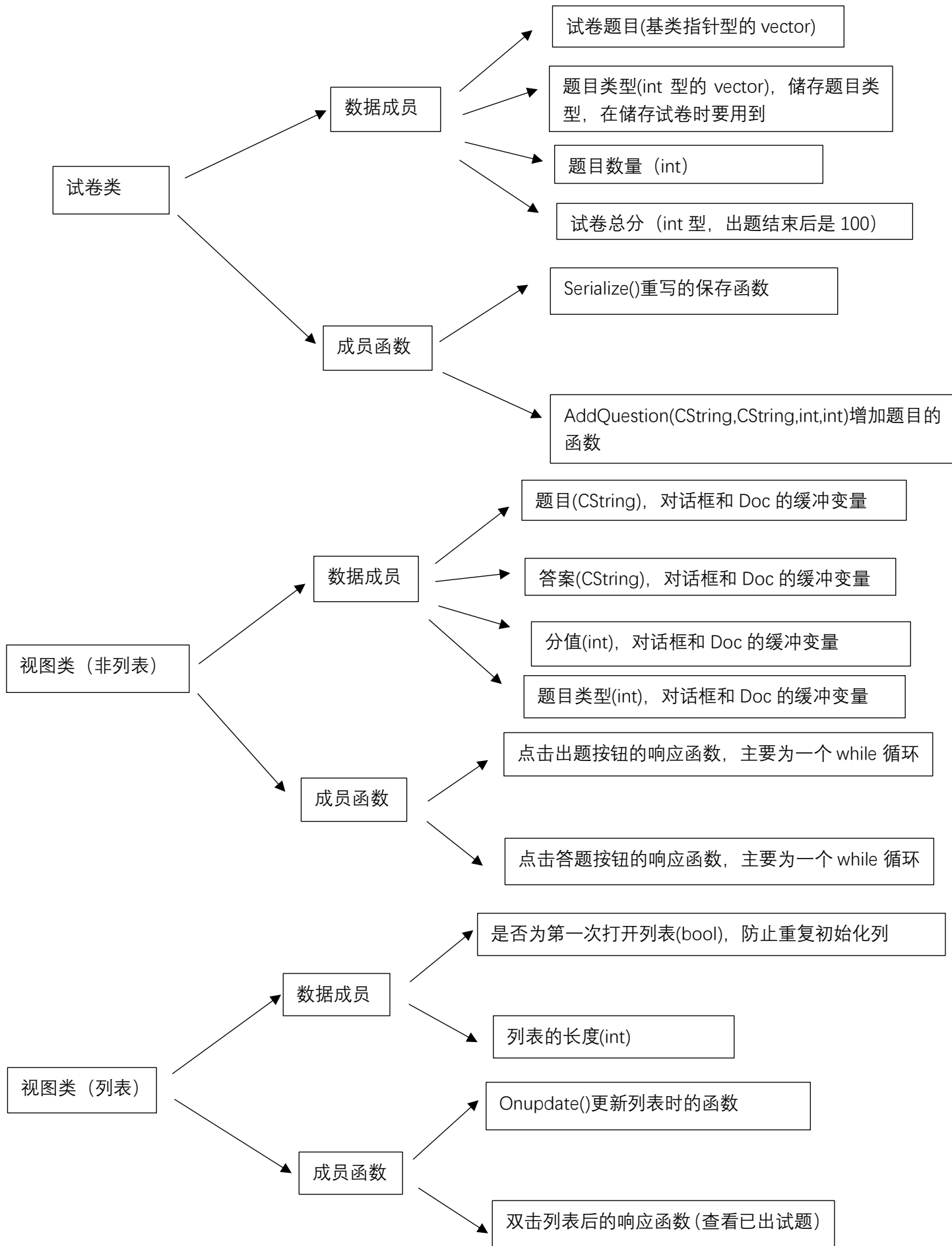
2) 点击答题按钮开始答题，触发 View 中的响应函数，同样是一个循环，不断将用户的答案储存下来并放到 View 视图中。

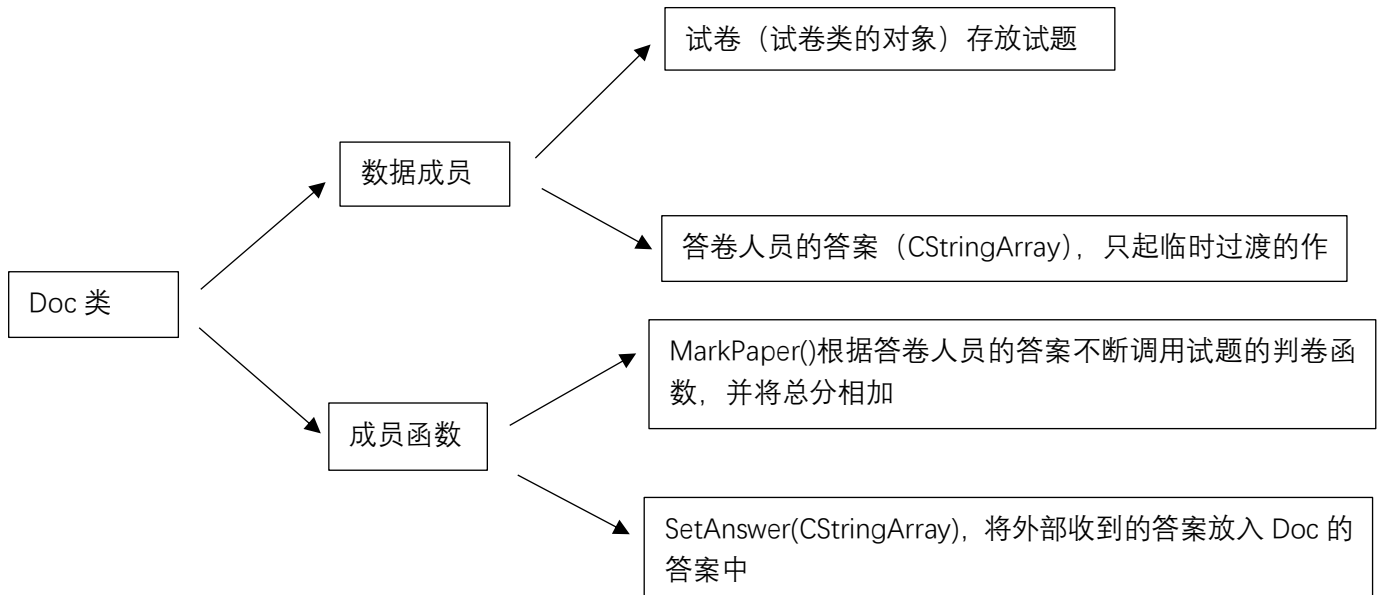
3) 点击交卷按钮后结束循环，将答题人员的答案传入 Doc 中，调用里面的判卷函数，最后计算出总分后传出，利用对话框显示给用户。

2.4 结构



注：所有题型的数据成员都是相同的，选择题的题目和选项都放在题目变量中，用";"隔开，同理多选题和简答题的答案都放在答案变量中，用";"隔开。





注：对话框类基本上都是对应的编辑框和对应的变量相关联，在确定按钮中设置了边界检查，接受数据之后将数据通过数据成员传到 View 类中，再传入 Doc 中，在此不详细列出

2.5 功能需求与程序的关系

数据储存全部再 Doc 中，出题人员通过 View 函数中的循环语句不断与出题对话框交互，将数据放入 Doc 中，答题人员同样同过 View 函数中的循环语句与对话框交互，并将答案传入 Doc 中，Doc 完成数据的收集和处理后给出总分，再 View 中通过对话框弹出总分。

2.6 人工处理过程

人工操作只涉及出题和答题，核算分数和边界检查等过程全部由系统完成，保证了运行的准确性。

3.接口设计

3.1 用户接口

用户的接口限于菜单中出题、答题的按钮，做到了 least privilege，保证不会因为用户的错误操作影响系统的运行。

3.2 外部接口

外部接口包括储存和打开，通过与系统硬盘的交互达到数据保存和打开的效果。

3.3 内部接口

模块间接口采用数据耦合方式，通过参数表达传递数据，交换信息。

4.运行设计

4.1 运行模块结合

具体软件的运行模块组合为文档-视图-对话框，各个模块在软件运行过程中能较好的交换信息。

4.2 运行控制

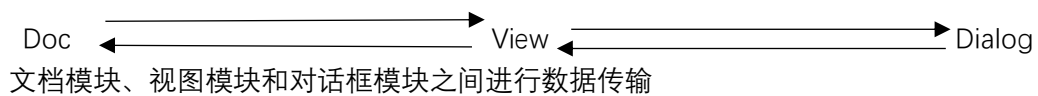
软件具有较友好的界面，基本能够实现用户的数据处理要求。

4.3 运行时间

系统的运行时间基本可以达到用户所提出的要求。

5.系统数据结构设计

5.1 数据结构要点



6.系统出错处理设计

6.1 出错信息

系统在运行过程中很可能出现因题目填写不完全、分值填写不正确、总分溢出等情况。

6.2 补救措施

在重要的数据处理节点添加条件判断，如果数据合法允许用户进行下一步操作，数据非法则弹出对话框说明错误并拒绝执行下一步，确保了程序的准确性。