Windows程序设计实验报告

实验序号：2

题目： 界面UI设计(1)

内容： 儿童算数作业生成器

小组成员：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 5120173574 | 班级 | 信安1704 |
| 姓名 | 杨欣 |  |  |

成绩：

评语：

日期： 2019/11/24

**设计内容、方法与难点：**

本课题设计的内容包括了自动生成加法计算题，可以设置加法计算题数量、位数类型（一位数或两位数）、计算类型（加法、减法、乘法）；自动统计每次作业的成绩包括（题目总数、运算正确题数、错误题数、未完成题数、时间）；自动存储每次成绩结果。

具体采用的方法是建立三个对话框，在对话框类里面进行操作，以及将数据存储到Doc类里面。

课题的难点有两个方面：

1. 数据的交互。该问题通过对View类发送消息，以及View类对Doc类发送消息解决的。
2. 对话框上图片的加载。使用了Picture控件以及为位图解决的。
3. 对话框上树状图的加载。使用了Tree控件解决的。
4. 对话框上表格的加载。使用了List控件解决的。
5. 界面初始化时的按钮。使用了CButton，在MainFrm的OnCreate里面动态创建的按钮。

**作品特色**

本作品的特点有：

1. 界面美观，操作简单。
2. 以一天的时间为单位存储做题记录。
3. 无闪烁，系统不会崩溃。

# **项目类设计：**

|  |
| --- |
| CGenerateQuestion：  m\_problems：生成的问题  m\_digit：位数类型     m\_type：计算 类型  m\_warning：输入错误报错  OnClickedConfirmBnt()：作业添加完毕  OnBnClickedAdd()：添加作业  init()：初始化  generateProblem(int num):生成作业  CQuestion：  m\_x：第一个参数  m\_y：第二个参数   m\_type：计算类型  m\_result：输入的答案   m\_correctResult： 正确答案  m\_checkResult：结果  m\_allNumber： 问题总数  m\_alreadyNumber： 已做题总数  m\_index： 当前做题序号  OnClickedQuestionNext()：下一题  OnInitDialog()：初始化  OnBnClickedQueationPrevious()：上一题  OnBnClickedQuestionSubmit()：提交  OnBnClickedButton4()：取消答题  init(vector<int> \*pro)：初始化  repaint()：重新写题目  CHistory：  m\_fileNames：文件总记录   m\_x： 题目第一个参数  m\_type：题目运算类型  m\_y： 题目第二个参数  m\_res： 输入的答案   m\_judgeRes：判断结果   m\_correctRes： 正确答案  m\_historyList：题目列表  m\_timeTree：历史文件树状图  m\_number1：题目第一个参数的十位数图片  m\_number2：题目第一个参数的个位数图片  m\_number3：题目第二个参数的前一位图片  m\_number4：题目第二个参数后一位图片  m\_queType：题目  init()：初始化，设置树状图和列表  OnInitDialog()： 初始化  getAllFile()：获取所有历史文件  getHistory(int index)： 获取某个历史文件的题目  drawHistory()： 将题目写入表格  drawQuestion(int index)：将当前题目画出来  OnSelchangedFileTree(NMHDR \*pNMHDR, LRESULT \*pResult)： 点击树状图时运行函数  OnDblclkHistoryList(NMHDR \*pNMHDR, LRESULT \*pResult)：点击表格时运行函数  setBitMap(int index1, int index2)：设置图片 |

# **（要求有类图和类功能、成员函数和成员变量的文字介绍）**

# **重点函数简介**

（要求函数必须有注释和说明）

|  |
| --- |
| void CQuestion::repaint(){  CString str1,str2,str;  int x,y,type;  //如果是第一题就取全局变量，因为第一题的数据存储到FIRST变量中  if(m\_index == 0 && m\_result.size() == 0){  x = FIRST[0];  y = FIRST[2];  type = FIRST[1];  //设置文本  str.Format(\_T("0/%d"), FIRST[3]/3);  SetDlgItemText(IDC\_QUESTION\_RATE, str);  }else{  x = m\_x[m\_index];  y = m\_y[m\_index];  type = m\_type[m\_index];  //设置文本  str.Format(\_T("%d/%d"),m\_alreadyNumber,m\_allNumber);  SetDlgItemText(IDC\_QUESTION\_RATE, str);  }  if(type == 0){  str.Format(\_T("%d\*%d"),x,y);  }else if(type ==1){  str.Format(\_T("%d+%d"),x,y);  }else{  str.Format(\_T("%d-%d"),x,y);  }  //设置问题文本  SetDlgItemText(IDC\_QUESTION\_CONTENT, str);  //设置答案框  if(m\_result[m\_index]!=-10001){  str.Format(\_T("%d"),m\_result[m\_index]);  SetDlgItemText(IDC\_QUESTION\_RESULT, str);  }else{  //如果结果是-10001，表示这道题还未出入答案  SetDlgItemText(IDC\_QUESTION\_RESULT, \_T(""));  }  }  void CQuestion::OnBnClickedQueationPrevious()  {  // TODO: 在此添加控件通知处理程序代码  CString answer;  GetDlgItemText(IDC\_QUESTION\_RESULT, answer);  //获取输入框答案  int number;  //如果没有输入值，就存储为-10001  if(answer == ""){  number = -10001;  //如果没有输入值，而之前的答案有值，那么已做题减一  if(m\_result[m\_index] != -10001){  m\_alreadyNumber--;  }  }else {  number = \_ttoi(answer);  //如果有输入值，而之前的答案没有值，那么已做题数加一  if(m\_result[m\_index] == -10001){  m\_alreadyNumber++;  }  }  //输入框的值存入数组  m\_result[m\_index] = number;  //翻页  if(m\_index <= 0){  AfxMessageBox(\_T("已经是第一题啦！"));  }else{  m\_index--;  repaint();  }  }  void CGenerateProblem::generateProblem(int num){  //获取运算位数  int digit = m\_digit.GetCurSel();  //获取运算类型  int type = m\_type.GetCurSel();  int range = 0;  //如果运算位数是1，那么就生成1~10以内的数字，否则生成10~100以内的数字  if(digit == 1){  range = 10;  }else{  range = 100;  }  //每次重启后的随机数都不一样  srand(time(NULL));  //生成随机数，m\_problems存储的是运算的两个数字和运算类型  for(int i=0;i<num;i++){  int x = range / 10 + rand()%range;  m\_problems.push\_back(x);  m\_problems.push\_back(type);  int y = range / 10 + rand()%range;  m\_problems.push\_back(y);  }  CString str;  //设置总题数文本  str.Format(\_T("%d"), m\_problems.size()/3);  SetDlgItemText(IDC\_ALL\_NUMBER, str);  SetDlgItemText(IDC\_PROBLEM\_NUMBER, \_T(""));  }  void CHistory::drawQuestion(int index)  {  //清除Picture控件的图片  m\_number1.ShowWindow(FALSE);  m\_number2.ShowWindow(FALSE);  m\_number3.ShowWindow(FALSE);  m\_number4.ShowWindow(FALSE);  m\_queType.ShowWindow(FALSE);  //获取第一个数字的前后两位  int a = m\_x[index] / 10;  int b = m\_x[index] % 10;  //如果前一个数字不为0，则表示这个为两位数，那么就设置其图片  if(a != 0){  setBitMap(1,a);  }  //设置后一个数字图片  setBitMap(2,b);  a = m\_y[index] / 10;  b = m\_y[index] % 10;  //如果第二个数字是两位数，那么就设置两个图片  if(a!=0){  setBitMap(3, a);  setBitMap(4, b);  }else {  //如果不是两位数，那么就将数字设置到前一个图片位  setBitMap(3, b);  }  if(m\_type[index] == 0){//如果运算位为乘  setBitMap(5, 10);  }else if(m\_type[index] == 1){//如果运算位为减  setBitMap(5, 12);  }else {//如果运算位为加  setBitMap(5, 11);  }  //设置正确答案  CString corrent;  corrent.Format(\_T("%d"), m\_correctRes[index]);  SetDlgItemText(IDC\_HISTORY\_CORRECT, corrent);  //设置结果  CString res;  res.Format(\_T("%d"), m\_res[index]);  //如果答案是-10001，那么这道题就是没有回答的  if(m\_res[index] == -10001){  SetDlgItemText(IDC\_HISTORY\_RESULT, \_T("未回答"));    }else{  SetDlgItemText(IDC\_HISTORY\_RESULT, res);  }  }  void CHistory::getHistory(int index)  {  CString name(\_T("..\\HISTORY\\"));  name += m\_fileNames[index];  name += ".txt";  CFileException fx;  CStdioFile file;  //获取文件  bool flag = file.Open(name, CFile::typeText|CFile::modeCreate|CFile::modeReadWrite|CFile::modeNoTruncate, &fx);  if(!flag){  AfxMessageBox(\_T("文件打开失败！请重试！"));  return ;  }  CString str(\_T(""));  CString num;  int number;  //清除所有数据  m\_x.clear();  m\_type.clear();  m\_y.clear();  m\_res.clear();  m\_judgeRes.clear();  m\_correctRes.clear();  int index1 = 0;  //开始读文件数据  while(file.ReadString(str)){  int j = 0,i;  //找第一个数字  //从第j位开始找间断符所在位置，返回空格符下标  i = str.Find(\_T(" "),j);  //从第j位开始，截取i-j的长度  num = str.Mid(j, i - j);  //CString转为int  number = \_ttoi(num);  m\_x.push\_back(number);  //下一次从空格符的下一位开始寻找  j += (i - j + 1);  //找运算类型  i = str.Find(\_T(" "),j);  num = str.Mid(j, i - j);  number = \_ttoi(num);  m\_type.push\_back(number);  j += (i - j + 1);  //找第二个数字  i = str.Find(\_T(" "),j);  num = str.Mid(j, i - j);  number = \_ttoi(num);  m\_y.push\_back(number);  j += (i - j + 1);  //找输入结果  i = str.Find(\_T(" "),j);  num = str.Mid(j, i - j);  number = \_ttoi(num);  m\_res.push\_back(number);  j += (i - j + 1);    int correctllyRes = 0;  //算出正确答案  if(m\_type[index1] == 0){  correctllyRes = m\_x[index1]\*m\_y[index1];  }else if(m\_type[index1] ==1){  correctllyRes = m\_x[index1]+m\_y[index1];  }else{  correctllyRes = m\_x[index1]-m\_y[index1];  }  //存储正确答案  m\_correctRes.push\_back(correctllyRes);  //判断结果  if(m\_res[index1] == -10001){  m\_judgeRes.push\_back(-1);  }else if(m\_res[index1] == correctllyRes){  m\_judgeRes.push\_back(1);  }else{  m\_judgeRes.push\_back(0);  }  //当前行数加一  index1++;  }  //关闭文件  file.Close();  } |

# **运行结果**

程序运行界面如下。界面显示信息清楚，内容包括：系统初始界面、题目生成界面、做题界面等



图 1 系统初始界面

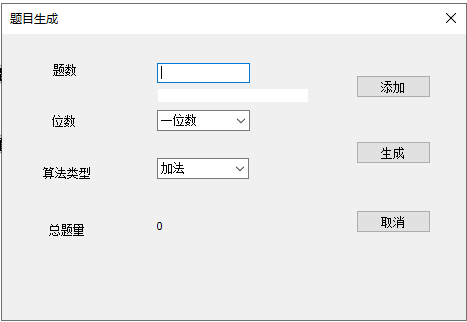


图 2 题目生成界面



图 3 做题界面



图 4 做题记录界面

# **有待解决的问题**

1. 界面不够卡通
2. 代码太杂乱
3. 只向Doc类传输数据，没有获取数据