

四则运算器 API 手册

by 第四组

类的说明

类名：Core

成员函数：

函数名	说明	默认值
<code>int get_que_num()</code>	返回题目数量 <code>que_num</code> ;	10
<code>int get_data_num()</code>	返回操作数个数 <code>data_num</code> ;	5
<code>int get_range()</code>	返回操作数的范围 <code>range</code> ;	10
<code>int get_opr_type()</code>	返回运算符种类 <code>opr_type</code> ; 0 : 加减, 1 : 加减乘, 2 : 加减乘除 , 3 : 加减乘除、乘方;	0 (加减模式)
<code>int get_data_type()</code>	返回操作数的种类 <code>data_type</code> ; 0 : 整数 , 1 : 小数, 2 : 分数;	0 (整数模式)
<code>int get_accuracy()</code>	返回小数模式下精度 <code>accuracy</code> ;	2 (保留两位小数)

成员函数：

函数名	说明	默认值
void set_que_num(int a)	设置题目数为 a 道 (a > 0) ；	10
void set_data_num(int a)	设置操作数为 a 个 (a > 0) ；	5
void set_range(int a)	设置操作数的范围 0 ~ a ；	10
void set_opr_type(int a)	设置运算符种类； a = 0 加减， a = 1 加减乘， a = 2 加减乘除， a = 3 加减乘除乘方；	0 (加减模式)
void set_data_type(int a)	设置操作数的种类； a = 0 整数， a = 1 小数， a = 2 分数；	0 (整数模式)
void set_accuracy(int a)	设置小数模式下操作数精度； 保留 a 位小数， a > 0；	2 (保留两位小数)
string *getQue ()	返回 string 数组首地址 Que； Que [i] 即为第 i 道题， 1 <= i <= que_num；	&" "
string *getAns ()	返回 string 数组首地址 Ans； Ans[i] 即为第 i 道题答案， 1 <= i <= que_num；	&" "

调用实例

```
1 int main()
2 {
3     Core cal;        //根据类 Core 创建对象 cal
4     string *q, *a;    //q a 分别为题目和答案 string 数组首地址
5
6     cal.set_data_num(10);    //设置操作数为 10 个
7     cal.set_que_num(100);    //设置题目数为 100 道
8     cal.set_range(10);      //设置操作数范围 0~10
9     cal.set_opr_type(3);     //设置运算符类型，参数详见上
10    cal.set_data_type(1);     //设置操作数种类，参数详见上
11    cal.set_accuracy(4);      //设置小数模式下的精度
12
13    q = cal.getQue();         //获取题目 string 数组指针，详见上
14    a = cal.getAns();         //获取题目答案 string 数组指针，详见上
15    for( int i = 1; i <= cal.get_que_num(); i++) //打印
16        cout << q[i] << endl << "=" << a[i] << endl << endl;
17    system("pause");
18    return 0;
19 }
```

DLL 使用步骤

1. 将 CORE_DLL.dll, CORE_DLL.lib 以及 targetver.h 三个文件复制到存放即将调用 core 的 .cpp 的文件夹中;
2. 在资源文件中 添加 现有项 "CORE_DLL.lib";
3. 在头文件中添加 现有项 "targetver.h" 并引用;

注意事项

1. 设置入口做取模处理, 尽量不要非法设置;
2. 分数显示均采用 top | bottom 形式, 例如 3 | 5、6 | 2;
3. 返回的 string 数组指针, string 下标从 1 开始, 题号与答案号对应;
4. 重新生成一套题不需要新建对象, 只需重新调用 getQue ();
5. 注意 x64 和 x86 不同的 DLL 应用;

By 第四组

范力 邹卫其