|  |
| --- |
| using namespace std; |
| /\* Создадим абстрактный класс обработки ошибок, просто для примера \*/ |
| class Exception |
| { |
| protected: |
| int num; |
| char\* str; |
| public: |
| virtual void display () = 0; |
| }; |
| /\* Некий глобальный класс для обработки ошибок \*/ |
| class OtherException : public Exception |
| { |
| public: |
| OtherException (int num, char\* str) |
| { |
| this->num = num; |
| this->str = str; |
| } |
|  |
| void display () |
| { |
| cout << "OtherException! - #" << num << ", " << str << endl; |
| } |
| }; |
|  |
| class Math  { |
| /\* Класс обработки ошибок только в классе Math\*/ |
| class ExceptionMath : public Exception |
| { |
| public: |
| ExceptionMath (int num, char\* str) |
| { |
| this->num = num; |
| this->str = str; |
| } |
| void display () |
| { |
| cout << "Exception! - #" << num << ", " << str << endl; |
| } |
| }; |
| public: |
| float Division (float v1, float v2); |
| int Multiplication (int v1, int v2); |
| }; |
| float Math::Division(float v1, float v2){ |
| try |
| { |
| /\* Создается объект класса ExceptionMath, который передается в блок catch \*/ |
| if (v2 == 0) |
| throw ExceptionMath(0, (char\*)"Division by zero"); |
|  |
| / \* Создается объект класса OtherException, который передается в блок catch \*/ |
| if (v2 == 666 || v1 == 666) |
| throw OtherException(333, (char\*)"Are you sure you want to division of the devil?"); |
| return v1 / v2; |
| } |
| /\* Блок ниже принимает исключения только типа ExceptionMath \*/ |
| catch (ExceptionMath exp) |
| { |
| exp.display(); |
| return (float)NULL; |
| } |
|  |
| \* Блок ниже принимает исключения только типа OtherException \*/ |
| catch (OtherException exp) |
| { |
| exp.display(); |
| return (float)NULL; |
| } |
| } |
|  |
| int Math::Multiplication(int v1, int v2) |
| { |
| try |
| { |
| if (v2 == 1 || v1 == 1) |
| throw ExceptionMath(1, (char\*)"What fool would be multiplied by one?"); |
| if (v2 == 666 || v1 == 666) |
| throw OtherException(333, (char\*)"Are you sure you want to multiply the number of the devil?"); |
| return v1 \* v2; |
| } |
| catch (ExceptionMath exp) |
| { |
| exp.display(); |
| return (int)NULL; |
| } |
| catch (OtherException exp) |
| { |
| exp.display(); |
| return (float)NULL; |
| } |
| } |
|  |
| int main() |
| { |
| Math m; |
|  |
| cout << m.Multiplication(10, 15) << endl; //150 |
| cout << m.Division(10, 15) << endl; //0.6(6) |
|  |
| cout << m.Multiplication(10, 1) << endl; //Exception! - #1, |
| cout << m.Multiplication(1, 55) << endl; //Exception! - #1, |
| cout << m.Division(10, 0) << endl; //Exception! - #0, Division by zero |
|  |
| cout << m.Multiplication(10, 666) << endl; //OtherException! - #333 |
| cout << m.Division(5519, 666) << endl; //OtherException! - #333, |
| } |