int \_\_stdcall Factorial(int number)
  
{
  
 if (number < 0)
  
 {
  
 throw std::invalid\_argument("负数不能进行阶乘");
  
 }
  
 if (number == 0 || number == 1)
  
 {
  
 return 1;
  
 }
  
  
 // 检查乘法前是否会溢出
  
 int result = Factorial(number - 1);
  
 if (result > INT\_MAX / number)
  
 {
  
 throw std::overflow\_error("阶乘结果超出了int的范围");
  
 }
  
  
 return number \* result;
  
}

private void button1\_Click(object sender, EventArgs e)
  
{
  
 try
  
 {
  
 if (button1.Text == "计算阶乘")
  
 {
  
 int number = Convert.ToInt32(textBox1.Text); // 确保输入能转换为整数
  
 int result = NativeMethods.Factorial(number);
  
 labelResult.Text = $"阶乘结果: {result}";
  
 }
  
 else if (button1.Text == "计算数据差值")
  
 {
  
 int a = Convert.ToInt32(textBox1.Text); // 确保输入能转换为整数
  
 int b = Convert.ToInt32(textBox2.Text); // 确保输入能转换为整数
  
  
 int c\_result = NativeMethods.CalculateDifference(a, b);
  
 labelResult.Text = $"数据差值: {c\_result}";
  
 }
  
 }
  
 catch (FormatException fe)
  
 {
  
 MessageBox.Show($"输入格式错误: {fe.Message}");
  
 }
  
 catch (OverflowException oe)
  
 {
  
 MessageBox.Show($"输入数值过大或过小: {oe.Message}");
  
 }
  
 catch (Exception ex)
  
 {
  
 MessageBox.Show($"发生错误: {ex.Message}");
  
 }
  
}

### 4. 创建WinForm窗口程序并调用COM组件

为了测试我们创建的 COM 控件，我们开发了一个名为 ExpressComClient 的 WinForm 应用程序。此应用程序具有直观的用户界面，包含以下元素：

* **两个文本框** (txtNumber1 和 txtNumber2)：用于输入要进行运算的数字。
* **两个按钮** (btnMinus 和 btnDivide)：分别用于执行减法和除法运算。
* **一个标签** (lblResult)：用于显示运算结果。

在实现中，我们首先使用 Activator.CreateInstance 和 Type.GetTypeFromCLSID 方法动态创建 COM 组件的实例。然后在按钮的点击事件处理程序中调用 COM 组件的方法。

#### 实例化 COM 组件

首先，我们在表单的构造函数中初始化 COM 组件：

public partial class MainForm : Form  
{  
 private dynamic expressComObject;  
  
 public MainForm()  
 {  
 InitializeComponent();  
 InitializeComObject();  
 }  
  
 private void InitializeComObject()  
 {  
 Guid clsid = new Guid("6F304CC0-E3E2-48D6-A831-7C4CF269F3FB"); // 替换为COM组件的CLSID  
 Type comType = Type.GetTypeFromCLSID(clsid);  
 expressComObject = Activator.CreateInstance(comType);  
 }  
}

#### 按钮点击事件处理

在按钮的点击事件处理程序中，我们从文本框中获取输入的数字，然后使用 COM 组件的方法来执行相应的运算。

* **减法运算**：
* private void btnMinus\_Click(object sender, EventArgs e)  
  {  
   int number1 = Convert.ToInt32(txtNumber1.Text);  
   int number2 = Convert.ToInt32(txtNumber2.Text);  
   lblResult.Text = expressComObject.Minus(number1, number2);  
  }
* **除法运算**：
* private void btnDivide\_Click(object sender, EventArgs e)  
  {  
   int number1 = Convert.ToInt32(txtNumber1.Text);  
   int number2 = Convert.ToInt32(txtNumber2.Text);  
   lblResult.Text = expressComObject.Divide(number1, number2);  
  }

通过这种方式，我们可以在 WinForm 应用程序中动态调用 COM 组件，从而展示 COM 组件的功能和灵活性。

这个修改后的报告部分详细解释了如何在 WinForm 应用程序中创建和使用 COM 组件的实例，并展示了如何在用户界面与这些组件交互。这些修改应该使报告更加清晰和准确地反映了您的项目实现。