**《C#程序设计》课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **12021051064** |  | **姓名：** | **华心童** |

**实验十三、异常处理**

**一、实验目的**

1、了解异常处理结构。

2、了解异常文件处理。

3、将实验报告云盘，文件名是**学号姓名实验13**命名。

**二、实验内容**

实验准备：在F盘创建学号文件夹，用于存放实验结果文件。

运行VS2013，创建解决方案是“**Ex学号-13**”。

1. **基本实验**

**任务1、异常处理【项目名P1301】**

**要求：**完成两个整数的除法运算，并统计计算次数，超出定义的次数后，退出。

实现异常处理（包括格式异常、除零异常、范围异常、未知异常）。

**程序源程序代码：**粘贴program.cs中代码

class Program

{

static void Main(string[] args)

{

bool flag = true;

int count = 0;

int result;

int[] s = new int[4];

while (flag)

{

Console.Write("除数：");

try

{

int a = int.Parse(Console.ReadLine());

result = 100 / a;

for (int i = 0; i < 6; i++)

s[i] = result;

}

catch (DivideByZeroException)

{

Console.WriteLine("分母不为0");

}

catch (FormatException)

{

Console.WriteLine("输入的格式不正确！");

}

catch (IndexOutOfRangeException)

{

Console.WriteLine("咦，超出范围了！");

}

catch (OverflowException)

{

Console.WriteLine("数据溢出");

}

finally

{

Console.WriteLine("一共计算了{0}次", ++count);

if (count > 4)

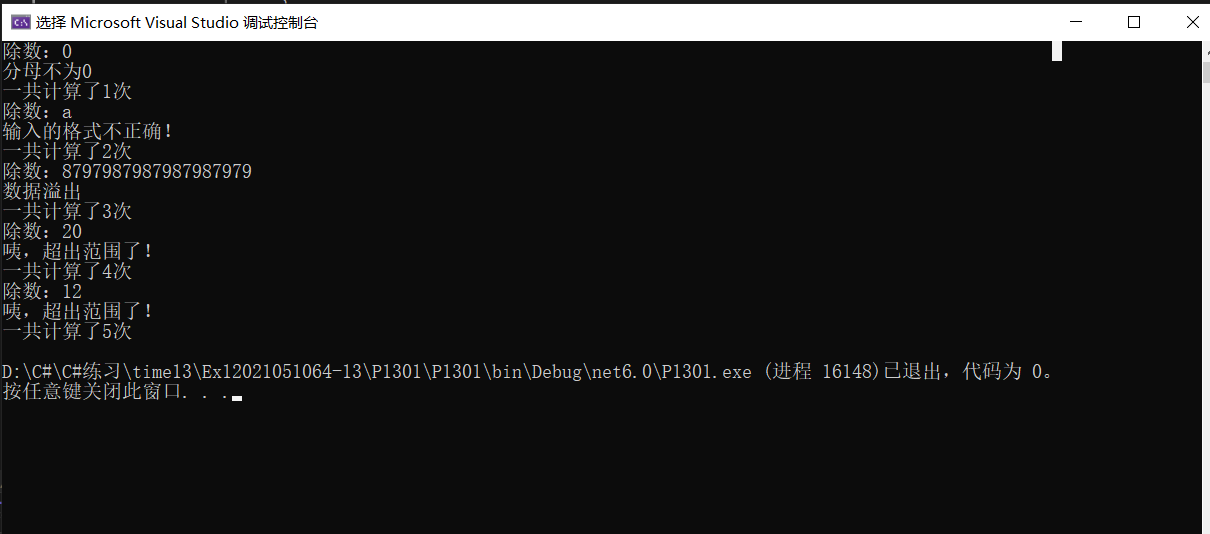
flag = false;

}

}

}

}//粘贴运行结果界面截图



**任务2、异常处理【项目名P1302】**

要求：定义一个数组，依次输入分数（要求必须是0-100之间），计算以QUIT结束。

统计：输入总数、正确的个数、总分、平均分。

异常处理：输入非数字、自定义异常类（从Exception类MyException派生，条件是分数不是0-100之间）、数组超出范围。

**程序源程序代码：**粘贴program.cs中代码

class MyException:Exception

{

public MyException(string message) : base(message)

{

}

}

class Program

{

static void Main(string[] args)

{

double[] c1 = new double[4];

double sum = 0;

double avgscore;

int count = 0;

bool flag = true;

int correct=0;

while (flag)

{

try

{

Console.Write("请输入{0}名同学的成绩：", c1.Length);

for (int i = 0; i < 5; i++)

{

c1[i] = double.Parse(Console.ReadLine());

correct++;

if (c1[i] >=0 && c1[i] <= 100)

count++;

if (c1[i] < 0 || c1[i]>100)

throw new MyException(string.Format("输入的分数不是0-100之间"));

sum += c1[i];

}

}

catch (FormatException)

{

Console.WriteLine("输入的格式不正确！");

}

catch (IndexOutOfRangeException)

{

Console.WriteLine("数组超出范围了！");

}

catch (MyException e)

{

Console.WriteLine(e.Message);

}

finally

{

if (sum > 0)

{

avgscore = sum / count;

Console.WriteLine("输入总数：{0}，正确的个数：{1}，总分：{2}，平均分：{3:F2}", correct, count, sum, avgscore);

sum = 0; correct = 0; count = 0;

}

if(Console.ReadLine().ToUpper()=="QUIT")

flag = false;

}

}

}

}

//粘贴运行结果界面截图



**任务3、异常处理【P1303】**

构建学生类。要求：输入学生的姓名，出生日期(8位字符)，和email地址，要求姓名不能为空串，年龄需要在18-45之间，email地址必须包含@字符。使用异常机制写出程序。

要求通过类的实例化，进行异常输出。

**参考代码：调试以下程序片段，使得程序能正确输出用户名及域名。**

string str = Console.ReadLine();//"abc@163.com"

int index = str.IndexOf("@");

string name = str.substring(0, index-1);

int dotIndex = str.IndexOf(".");

string domain = str.Substring(index + 1, dotIndex - index);

Console.WriteLine(name);

Console.WriteLine(domain);

**程序源程序代码：**粘贴program.cs中代码

class Student

{

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

private string birth;

public string Birth

{

get { return birth; }

set { birth = value; }

}

private string email;

public string Email

{

get { return email; }

set { email = value; }

}

public Student ()

{

}

public void Show()

{

Console.Write("输入学生姓名：");

string str = Console.ReadLine();

Console.Write("输入学生的出生日期：");

string birth = Console.ReadLine();

Console.Write("输入学生的email地址：");

string email = Console.ReadLine();

int index = email.IndexOf("@");

string name = str.Substring(0, str.Length);

int old;

DateTime dt1 = DateTime.Now;

string year = (birth.Substring(0, 8));

IFormatProvider format = new System.Globalization.CultureInfo("zh-CN");

string TarStr = "yyyyMMdd";

DateTime dt2 = DateTime.ParseExact(year, TarStr, format);

birth = string.Format("该客户出生日期：{0:yyyy年MM月dd日}", dt2);

Console.WriteLine(birth);

old = dt1.Year - dt2.Year;

Console.WriteLine("学生的姓名：{0},年龄：{1},学生的邮箱：{2}", name, old, email);

int lastindex = email.IndexOf("@");

int dotIndex = email.IndexOf('.');

int dotLastIndex = email.IndexOf('.');

string domain = email.Substring(index, dotIndex - index);

if (name==null||name.Length == 0)

throw new ArgumentNullException("姓名不能为空串");

if (old < 18 || old > 45)

throw new ArgumentOutOfRangeException("异常：年龄必须在18-45之间");

if (index == -1 || index == 0 || lastindex == email.Length - 1 || lastindex != index || dotIndex == -1 || dotIndex != dotLastIndex || dotIndex == 0 || dotLastIndex == email.Length - 1)

{

throw new FormatException(String.Format("这不是正确的Email地址"));

}

else Console.WriteLine(domain);

}

}

class Program

{

static void Main(string[] args)

{

try

{

Student s1 = new Student();

s1.Show();

}

catch (ArgumentNullException)

{

Console.WriteLine("Exception：姓名不能为空串");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Exception：年龄必须要在18-45之间");

}

catch (FormatException)

{

Console.WriteLine("Exception：这不是正确的Email地址");

}

try

{

Student s2 = new Student();

s2.Show();

}

catch (ArgumentNullException)

{

Console.WriteLine("Exception：姓名不能为空串");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Exception：年龄必须要在18-45之间");

}

catch (FormatException)

{

Console.WriteLine("Exception：这不是正确的Email地址");

}

try

{

Student s3 = new Student();

s3.Show();

}

catch (ArgumentNullException)

{

Console.WriteLine("Exception：姓名不能为空串");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Exception：年龄必须要在18-45之间");

}

catch (FormatException)

{

Console.WriteLine("Exception：这不是正确的Email地址");

}

try

{

Student s4 = new Student();

s4.Show();

}

catch (ArgumentNullException)

{

Console.WriteLine("Exception：姓名不能为空串");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Exception：年龄必须要在18-45之间");

}

catch (FormatException)

{

Console.WriteLine("Exception：这不是正确的Email地址");

}

}

}

//粘贴运行结果界面截图



