**实验报告**

**学院（系）名称：**计算机科学与工程学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 王帆 | **学号** | 20152180 | **专业** | 计算机科学与技术 |
| **班级** | 2015级 1班 | **实验项目** | 实验一：C#的数据类型和控制结构 | | |
| **课程名称** | | . NET程序设计 | | **课程代码** | 0667066 |
| **实验时间** | | 2017年3月30日 12：45-15：45 | | **实验地点** | 7-215 |
| **批改意见** | | | | **成绩** |  |
| **教师签字：** | | | | | |
| 1. 实验目的 2. 熟悉Visual Studio 2010开发环境，掌握C#数据类型； 3. 掌握分支语句、循环语句、跳转语句及异常处理语句； 4. 掌握数组的使用； 5. 掌握面向对象的编程思想：类、接口等的使用； 6. 字符串的常用方法。 7. 实验环境  * 开发环境：PC机，Windows 7操作系统； * 开发工具：Visual Studio 2010以上。  1. 实验要求 2. 认真完成实验内容，编写程序代码； 3. 要求所有程序有测试数据，并输出实验的正确结果； 4. 书写并提交实验报告。 5. 实验内容 6. **理解C#程序运行机制**   通过C# Hello World！程序了解最基本的C#编译过程以及运行机制  **代码实现：**  using System;  Namespace com.tjut{  Class HelloWorld{  Static void Main(){  Console.WriteLine(“第一个C#程序!!”);  Console.Read();  }  }  }  **运行结果：**     1. **创建控制台应用程序，完成下列功能：**   要求定义不同类型的变量：整型、浮点类型、布尔类型、字符和字符串类型等，并进行赋值。   * 输出各变量的值； * 实现不同数据类型的转换；   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace EXP1  {  class Program  {  static void Main(string[] args)  {  int i=1;  double d=3.1415926;  bool b=true;  char c='c';  string str="string";  Console.WriteLine("下面执行输出变量:");  Console.WriteLine("int:" + i + "\ndouble:" + d + "\nbool:" + b + "\nchar:" + c + "\nstring:" + str+"\n");  Console.Write("请输入一串数字:");  string s=Console.ReadLine();  Int64 s\_i=Int64.Parse(s);    Console.WriteLine("\n下面执行数据类型转换:");  Console.WriteLine("double->int:" + (Int32)d);  Console.WriteLine("string->int（Type.Parse方法）:"+s\_i);  Console.WriteLine("string->int（Convert方法）:" + Convert.ToInt32(s));  }  }  }  **运行结果：**     1. **分支和循环语句** 2. 利用分支、循环语句及break或continue等实现1-100所有整数或奇数的和；   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace EXP2  {  class Program  {  static void Main()  {  Console.WriteLine("利用分支循环实现1-100内整数和与奇数和:");  int i = 0;  int sum = 0;  while (i < 100)  {  sum += ++i ;  }  Console.WriteLine("1+2+...+100="+sum);  for(i=1,sum=0;i<=100;i+=2)  {  sum+=i;  }  Console.WriteLine("1+3+...+99="+sum);  }  }  }  **运行结果：**     1. 利用switch—case实现成绩的判定   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace 成绩判定  {  class Program  {  static void Main(string[] args)  {  string score = Console.ReadLine();  switch (Convert.ToInt32(score)/10)  {  case 10:  case 9:  {  Console.WriteLine("优秀");  break;  }  case 8:  {  Console.WriteLine("良好");  break;  }  case 7:  {  Console.WriteLine("中等");  break;  }  case 6:  {  Console.WriteLine("及格");  break;  }  case 5:  {  Console.WriteLine("不及格");  break;  }  default:  {  Console.WriteLine("输入信息超过正常范围，退出");  break;  }  }    }  }  }  **运行结果：**   1. 异常处理程序：利用try-catch-finally语句实现异常程序的处理。   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace try\_catch\_finally  {  class Program  {  static void Main(string[] args)  {  int a, b;  try  {  Console.Write("请输入被除数:");  a = int.Parse(Console.ReadLine());  Console.Write("请输入除数:");  b = int.Parse(Console.ReadLine());  double sum = a /b;  //将b显式转换成double类型时，显示正无穷大？？？  Console.WriteLine(sum);  }  catch  {  Console.WriteLine("除数为0！");  }  finally  {  Console.WriteLine("Finally!");  }  Console.ReadLine();  }  }  }  **运行结果：**     1. 引用类型使用：如数组，体会引用类型的参数传递。   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace 引用实现  {  class Program  {  static void Main(string[] args)  {  int[] input = {1,2,3,4,5,6,7,8,9,10};  for (int i = 0; i<input.Length;i++ )  {  if ( input[i] % 7 == 0)  input[i] = 0;  Console.Write(input[i]+" ");  }  Console.ReadLine();  }  }  }  **运行结果：**     1. 类、接口及多态：   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace 类\_接口及多态  {  //定义接口Body  interface Body  {  float area();  }  //创建类Cube，实现Body接口  class Cube:Body  {  private float a;  public Cube(float a)  {  this.a = a;  }  public float area()  {  return 6 \* a \* a;  }  }  //创建类Ball，实现Body接口  class Ball : Body  {  private float r;  public Ball(float r)  {  this.r = r;  }  public float area()  {  return (float)(4 \* 3.14 \* r \* r);  }  }  class MainClass  {  public static float getArea(Body b)  {  return b.area();  }  static void Main(string[] args)  {  Cube cube = new Cube(10);  Ball ball = new Ball(5);  Console.WriteLine("Cube`s Area=" + getArea(cube));  Console.WriteLine("Ball`s Area=" + getArea(ball));  Console.ReadLine();  }  }  }  **运行结果：**     1. **重载：**   编写程序实现多个同名方法，方法功能实现：获得多个数据的和  **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace 重载  {  class Program  {  public static int sum(int a, int b)  {  return a + b;  }  public static int sum(int []a, int i)  {  int sum=0;  while (true)  {  if (i < a.Length)  sum += a[i];  else break;  i++;  }  return sum;  }  public static double sum(double a, double b)  {  return a + b;  }  public static double sum(double[] a, int i)  {  double sum = 0;  while (true)  {  if (i < a.Length)  sum += a[i];  else break;  i++;  }  return sum;  }  static void Main(string[] args)  {  int i\_a=1,i\_b=2;  int[] i\_arr = { 1, 2, 3, 4, 5, 6, 7 };  double d\_a = 1.5, d\_b = 2.7;  double[] d\_arr = { 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5 };  Console.WriteLine("重载开始：");  Console.WriteLine("(int)a+b="+sum(i\_a, i\_b));  Console.WriteLine("(double)a+b="+sum(d\_a, d\_b));  Console.WriteLine("(int)sum\_arr="+sum(i\_arr, 0));  Console.WriteLine("(double)sum\_arr=" + sum(d\_arr, 0));  }  }  }  **运行结果：**     1. 字符串：使用字符串的常用方法：Contains()、Split()及Join()等   **代码实现：**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace 字符串实现  {  class Program  {  static void Main(string[] args)  {  string str = "字符串实现";  Console.WriteLine(str);  Console.WriteLine("1.Contains方法：str中有“测试”二字吗 " + str.Contains("测试"));  string s = "I,am,a,student,from,China";  string[] temp = s.Split(new[] {','});  Console.WriteLine(s);  Console.WriteLine("2.Split方法：将单词换行分别输出： ");  foreach(string i in temp)  {  Console.WriteLine(i);  }  string result = string.Join("/", temp);  Console.WriteLine("3.Join方法：将单词加“/”输出： " + result);  }  }  }  **运行结果：**     1. 心得体会   通过本次实验，我加深了对C#语言的理解，主要是对于C#基本方法的实现与特殊特性的运用。本次实验中，最需要关注的问题是对于引用类型的使用，如数组的参数传递，它不同于C/C++中使用的指针类型，具有安全性能强，符合面向对象程序设计思想，以及便于使用等方面的优点。此外，本次实验是C#语言的基础实验，这有助于我们对于基本语句的巩固练习以及基本方法的掌握。本次实验的过程中出现了几个问题，首先是异常处理部分，由于对异常处理方法不够了解，导致我花费了较多的时间进行资料获取以及学习的过程，最后实现了对于除数为0情况下的异常处理。我认为程序语言的学习是一种触类旁通的过程，实际上异常处理也就是一些边界条件的处理，之前在其他课程中也涉及到很多，只是在C#中有了新的使用方法，因此理解的过程就并不复杂了，重要的是学会使用，并养成做异常处理的习惯。 | | | | | |