【思考题】

1. 环境变量path的作用。

答：提供windows命令行中指令的可执行文件路径，当我们在命令行中键入指令时，根据环境变量中的path值，找到对应的指令可执行文件进行执行。配置环境变量path后，使用命令行进行源程序编译时，可以不用到编译程序的根目录的bin下，可以在任意位置。

1. 如何在记事本和Visual Studio 2019中如何编写、编译和运行C#文件。

答：

①要想编译和运行通过记事本编写好的C#文件（例如上文的3-1.cs），需要在环境变量配置好的情况下在MS-DOS中输入以下命令：CSC 3-1.cs，然后敲入回车，cs文件进行编译并生产3-1.exe文件。在命令提示符窗口键入生产的可执行文件按回车键即开始执行程序并输出结果。

②通过VS2019左上角文件->新建->选择C#文件建立新的C#文件。输入代码，点击调试可以进行编译和运行。

# 实验一（2） C#基本语法编程练习

## 【实验目的】

1. 了解C#的数据类型
2. 掌握各种变量的声明方式
3. 理解运算符的优先级
4. 掌握c#基本数据类型、运算符与表达式的使用方法
5. 理解C#程序语法结构，掌握顺序结构、选择结构和循环结构语法的程序设计方法

## 【实验要求】

1. 写出程序，并调试程序，要给出测试数据和实验结果。
2. 整理上机步骤，总结经验和体会。
3. 完成实验日志和上交程序。

## 【实验内容】

1. 编写一个程序，输出两行结果：Hello，My name is “自己的姓名”, This is my first csharp program；我的C#学习之旅开始了。
2. 编写一个声明c#不同数据类型变量的程序
3. 编写一个使用运算符、表达式、变量的程序
4. 编写一个使用c#数据的程序
5. 编写表达式语句、复合语句的程序
6. 编写使用不同选择结构的程序
7. 编写使用不同循环结构的程序

### 一．编写程序，输出实验内容1要求的结果

参考代码如下：

#region Using directives

using System;

using System.Collections.Generic;

using System.Text;

#endregion

namespace Hello

{

class Hello

{

static void Main()

{

Console.WriteLine("Hello, My name is LiNing, This is my first csharp program");

Console.WriteLine("我的c#学习之旅开始了");

Console.ReadLine();

}

}

}

在最后将Console.ReadLine();行加上注释，看输出界面有何变化，掌握此语句控制命令操作符界面的使用技巧。

运行截图：

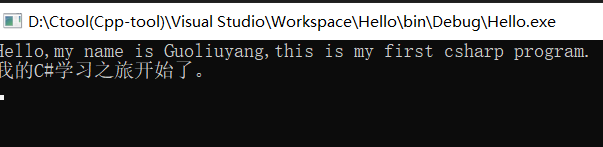


图1

如果将Console.ReadLine();行加上注释，则,不会打印两行字，程序很快结束了。

### 二．声明不同数据类型的变量

1．编写声明不同数据类型变量的程序文件datatype.cs，源代码如下:

#region Using directives

using System;

using System.Collections.Generic;

using System.Text;

#endregion

namespace DataType

{

class DataType

{

static void Main()

{

int myInt = 3;

short myShort = 32765;

uint myUint = 1;

float myFloat = 100.15f;

double myDouble = -99;

long myLong = 10000;

decimal myDecimal = -1.88m;

Console.WriteLine("myInt:{0},myShort:{1},myUint:{2},myFloat:{3}",myInt,myShort,myUint,myFloat);

Console.WriteLine("myDouble:{0},myLong:{1},myDecimal",myDouble,myLong,myDecimal);

Console.ReadLine();

}

}

}

1. 编译并运行该程序并修改各数据类型不同的数值，检验教材上所示数据范围。

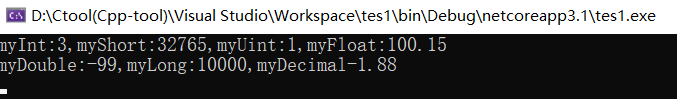


图2

1. 将两行中的WriteLine改为Write查看输出结果。

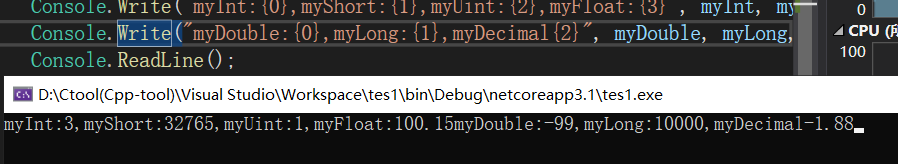


图3

4.分析以下程序功能：

short shortResult, shortVal = 4;

int integerVal = 67;

long longResult;

float floatVal = 10.5F;

double doubleResult, doubleVal = 99.999;

string stringResult, stringVal = "17";

bool boolVal = true;

Console.WriteLine("Variable Conversion Examples\n");

doubleResult = floatVal \* shortVal;

Console.WriteLine("Implicit, -> double: {0} \* {1} -> {2}", floatVal,

shortVal, doubleResult);

shortResult = (short)floatVal;

Console.WriteLine("Explicit, -> short: {0} -> {1}", floatVal,

shortResult);

stringResult = Convert.ToString(boolVal) +

Convert.ToString(doubleVal);

Console.WriteLine("Explicit, -> string: \"{0}\" + \"{1}\" -> {2}",

boolVal, doubleVal, stringResult);

longResult = integerVal + Convert.ToInt64(stringVal);

Console.WriteLine("Mixed, -> long: {0} + {1} -> {2}",

integerVal, stringVal, longResult);

Console.ReadLine();

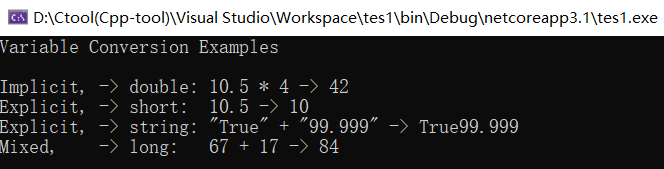


图4

该程序的功能是展示不同数据类型之间的运算和转换。

5.用enum定义字节类型的方位常量，打印出某一方位并将此方位值转化为字节类型，字符串类型值。分析输出结果的原因。回答以下问题：

Enum的缺省类型是什么？直接输出myDirection和(byte)myDirection有何区别。

class Variables

{

enum orientation : byte

{

north = 1,

south = 2,

east = 3,

west = 4

}

static void Main(string[] args)

{

byte directionByte;

string directionString;

orientation myDirection = orientation.north;

Console.WriteLine("myDirection = {0}", myDirection);

directionByte = (byte)myDirection;

directionString = Convert.ToString(myDirection);

Console.WriteLine("byte equivalent = {0}", directionByte);

Console.WriteLine("string equivalent = {0}", directionString);

Console.ReadLine();

}

}

### enum缺省类型是int；直接输出myDerection会输出枚举的成员，而输出(byte)myDerection会输出代表该成员的符号常量。

### 二．使用关系运算符和逻辑运算符

1．建立使用关系运算符和逻辑运算符的程序文件。Main方法中实例代码如下

static void Main(string[] args)

{

Console.WriteLine("Enter an integer:");

int myInt = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Integer less than 10? {0}", myInt < 10);

Console.WriteLine("Integer between 0 and 5? {0}",

(0 <= myInt) && (myInt <= 5));

Console.WriteLine("Bitwise AND of Integer and 10 = {0}", myInt & 10);

Console.ReadLine();

}

1. 编译运行该程序。并尝试myInt输入不同范围整数，非10和10时的输出差异。

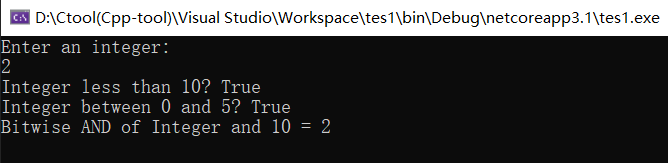


图5

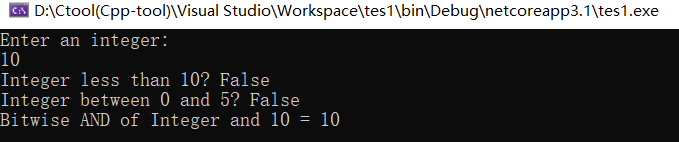


图6

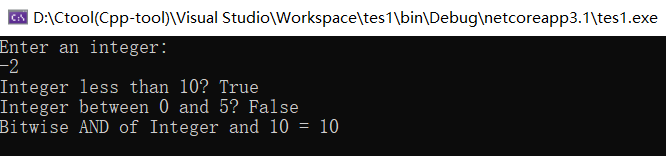


图7

### 三． 使用表达式语句与复合语句

1． 建立包含表达式语句程序，要求定义两个double型数据，从控制台输入你的名字和两个double数据，计算这两个数据的和、差、乘、除。源代码如下。

using System;

namespace Variables

{

class Variables

{

static void Main(string[] args)

{

double firstNumber, secondNumber;

string userName;

Console.WriteLine("Enter your name:");

userName = Console.ReadLine();

Console.WriteLine("Welcome {0}!", userName);

Console.WriteLine("Now give me a number:");

firstNumber = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Now give me another number:");

secondNumber = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("The sum of {0} and {1} is {2}.", firstNumber,

secondNumber, firstNumber + secondNumber);

Console.WriteLine("The result of subtracting {0} from {1} is {2}.",

secondNumber, firstNumber, firstNumber - secondNumber);

Console.WriteLine("The product of {0} and {1} is {2}.", firstNumber,

secondNumber, firstNumber \* secondNumber);

Console.WriteLine("The result of dividing {0} by {1} is {2}.",

firstNumber, secondNumber, firstNumber / secondNumber);

Console.WriteLine("The remainder after dividing {0} by {1} is {2}.",

firstNumber, secondNumber, firstNumber % secondNumber);

Console.ReadLine();

}

}

}

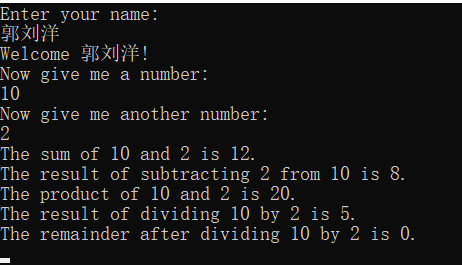


图8

2. 注意总结：Console.ReadLine方法，Convert.ToDouble方法的使用。

3. 查阅MSDN找出一系列转化成其它类型的Convert方法的使用。

### 四．使用选择语句

1.从键盘输入两个数进行比较，并定义一个字符串变量，当数1小于数2时，字符串变量为“less than”，当当数1等小于数2时字符串变量为“equal to”,当数1大于数2时字符串变量为“greater than”。

static void Main(string[] args)

{

string comparison;

Console.WriteLine("Enter a number:");

double var1 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Enter another number:");

double var2 = Convert.ToDouble(Console.ReadLine());

if (var1 < var2)

comparison = "less than";

else

{

if (var1 == var2)

comparison = "equal to";

else

comparison = "greater than";

}

Console.WriteLine("The first number is {0} the second number.",

comparison);

Console.ReadLine();

}

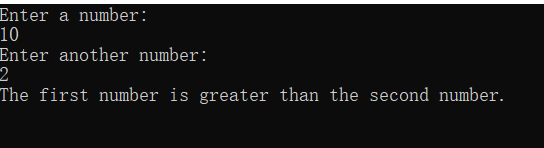


图9

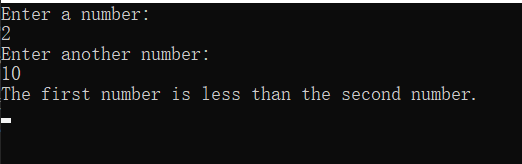


图10

2.定义三个常量字符串“karli”，"angelina"，"ploppy"，并从键盘输入一个名字，当名字与“karli”相同时输出我们的名字相同，当和"angelina"名字相同时输出你的名字太性感了，当和"ploppy"相同时输出这名字真傻。

static void Main(string[] args)

{

const string myName = "karli";

const string sexyName = "angelina";

const string sillyName = "ploppy";

string name;

Console.WriteLine("What is your name?");

name = Console.ReadLine();

switch (name.ToLower())

{

case myName:

Console.WriteLine("You have the same name as me!");

break;

case sexyName:

Console.WriteLine("My god, what a sexy name you have!");

break;

case sillyName:

Console.WriteLine("That's a very silly name.");

break;

}

Console.WriteLine("Hello {0}!", name);

Console.ReadLine();

}

### 

图11

### 五．使用循环语句

1． for 循环语句练习

（1） 程序功能要求：按5 度的增量打印出一个从摄氏温度到华氏温度的转换表。

static void Main(string[] args)

{

double Fa,Cel;

Cel =0;

for(Cel=0;Cel<100;Cel+=5)

{

Fa = Cel \* 9/5+32;

Console.WriteLine(Fa);

Console.ReadLine();

}

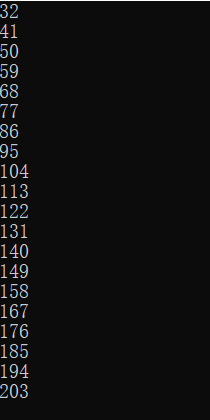


图12

（2）自行改造以上程序。

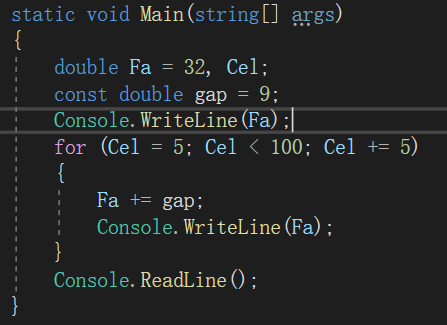


图13

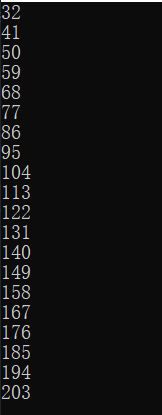


图14

2． while 循环语句练习

（1）程序功能要求：运行程序后从键盘输入数字1/2/3 后，可显示抽奖得到的奖品：恭喜你得了一辆汽车；不错啊，一台笔记本电脑；没白来，一个MP3；如果输入其它数字或字符显示“没有奖品给你!”。示例代码如下：

int choice;

choice = Convert.ToInt32(Console.ReadLine());

while (choice == 1)

{

Console.WriteLine("恭喜你得了一辆汽车");

break;

}

while (choice == 2)

{

Console.WriteLine("不错啊，一台笔记本电脑");

break;

}

while (choice == 3)

{

Console.WriteLine("没白来，一个MP3");

break;

}

while (choice != 1 && choice != 2 && choice != 3)

{

Console.WriteLine("没有奖品给你");

break;

}

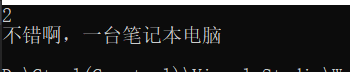


图15

（2）改造以上程序实现此功能；尝试将choice==1或2或3中的“==”改为一个“=”，看效果如何？并分析错误。

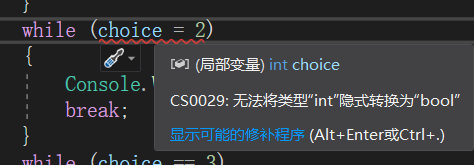


图16

错误有两个，一是图上提示的，无法隐式转换int成bool，而是若改为“=”，则括号中的表达式是确定的，不能让while语句进行判断。

3．do…while 循环语句练习

程序功能要求：输入你现有的存款和当前的年利率及你期望将来得到的存款，计算出存款多少年后才可以变成你期望的存款额。注意，若为一年输出year为year，若为多年输出year为years。参考代码如下：

static void Main(string[] args)

{

double balance, interestRate, targetBalance;

Console.WriteLine("What is your current balance?");

balance = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("What is your current annual interest rate (in %)?");

interestRate = 1 + Convert.ToDouble(Console.ReadLine()) / 100.0;

Console.WriteLine("What balance would you like to have?");

targetBalance = Convert.ToDouble(Console.ReadLine());

int totalYears = 0;

do

{

balance \*= interestRate;

++totalYears;

}

while (balance < targetBalance);

Console.WriteLine("In {0} year{1} you'll have a balance of {2}.",

totalYears, totalYears == 1 ? "" : "s", balance);

}

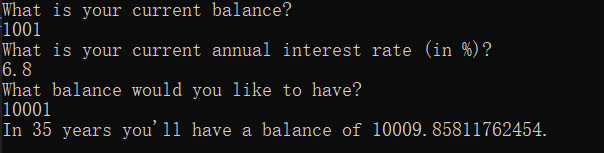


图16

【思考题】

1. while和do…while的区别。

答：

①语法结构不同。

②执行次数不同。while语句如果不满足循环条件，可以一次都不执行；do-while至少会执行一次循环体。

2．使用if...else 语句编写以下程序

（1）程序功能要求：使用if...else 语句构造多分支，判断某一年是否为闰年。闰年的条件是符合下面二者之一：能被4 整除，但不能被100 整除；能被4 整除，又能被100 整除。

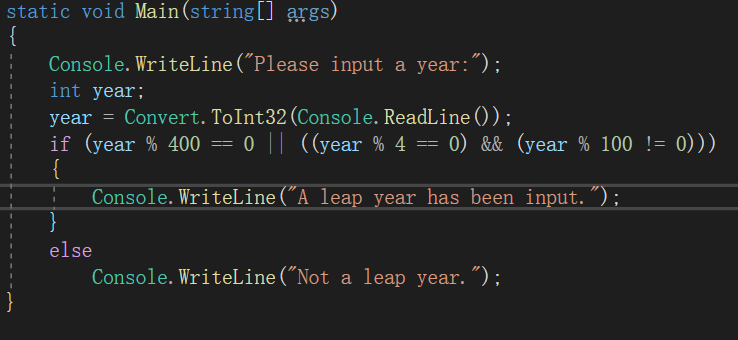


图17

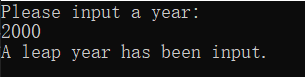


图18

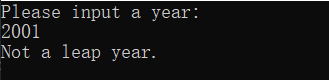


图19

3．使用switch 语句编写以下程序

（1）程序功能要求：在不同温度时显示不同的解释说明：有点冷，多穿衣服；正合适，出去玩吧；太热了，开空调。

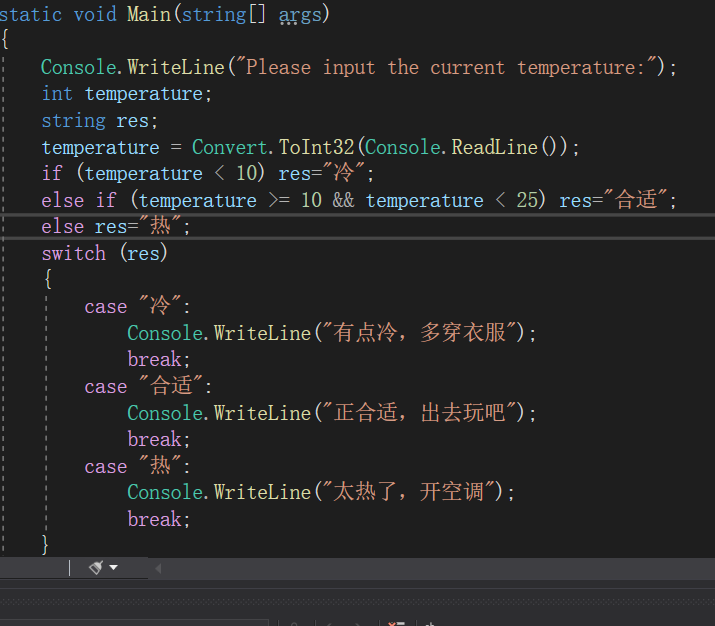


图20

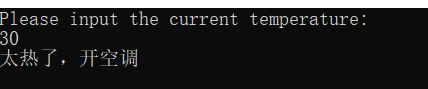


图21

4. 用do…while语句实现程序功能：求1＋2＋…+100 之和，并将求和表达式与所求的和显示出来。

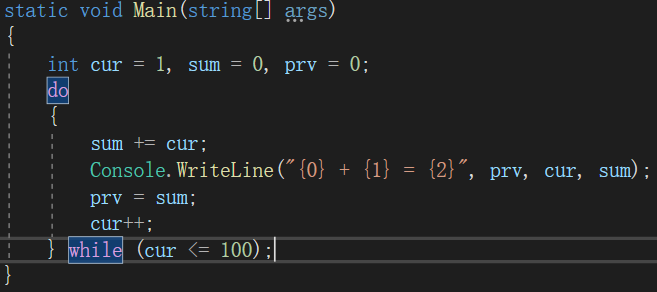


图22

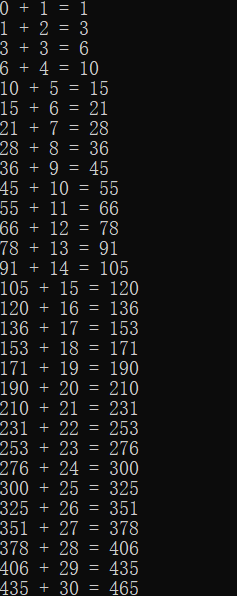


图23

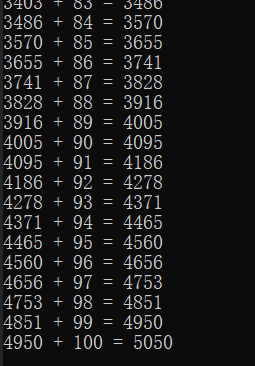


图24