

# 编程概述（.Net）课程知识点及软件编程热点分析

姓名：舒翔宇 学号：22021321062016

## 摘要

C#，一款由微软发布的面向对象的编程语言，是一款强大，运行高效，安全，稳定，语法体系先进的高级程序设计语言。对《编程概述(.Net)》的主要章节，输入输出，控制结构，数组，方法，文件等章节的主要知识点进行分析和理解，未来我们未来更深入的学习打下基础。目前软件行业发展空间巨大。移动互联网，物联网，计算机视觉等前沿技术的发展，软件更加智能化，充斥着我们的生活。我国对于软件产业高度重视，计算机技术，通信技术的发展，软件作为计算机应用的产品，拥有强大的力量，在方便我们的生活，提高社会生产效率，拉动经济增长起重要作用。

## 关键字

C#，主要的基础操作，软件产业，计算机前沿技术，发展趋势。

## 正文

### （一）编程概述(.Net)的主要知识点的分析与理解

本学期的《编程概述(.Net)》的主要学了五个章节的内容，输入输出，控制结构，数组，方法，文件这五个章节，本文将分别分析和理解这五个章节的重点知识，并举例说明。

#### 一）输入输出：

1) 输出操作。当我们需要在控制台显示一些提示或者显示结构时，C#中提供了两种输出的方式，“Console.Write();”和“Console.WriteLine();”。两者都能在控制台中输入，不同的是“Console.WriteLine();”将会回车换行，“Console.Write();”则不会。

输出是也有数据的格式化。

//例 1

源代码：

```
Console.WriteLine(“{0},{1}”,1,2);//输出的是 1, 2
Console.WriteLine(“{1},{0}”,1,2);//输出的是 2, 1
Console.WriteLine(“{0:N3}”,12);//显示小数点后三位, “12.000”
Console.WriteLine(“{0:8}”,21);//显示八位字符, “      12”
Console.WriteLine(“{0:-8}”,12); //左对齐, 显示八位字符, “12      ”
Console.WriteLine(“{0:C3}”,12); //显示金钱, “$12”
```

2) 输入操作。我们需要从控制台获取用户的输入的数据, 收集数据后返回给程序进行操作, C#提供了两种输入操作分别为“Console.Read();”和“Console.ReadLine();”。

“Console.ReadLine();”读取的是一行字符串, 返回 string 值, 按下 Enter 键时才停止输入数据的收集: “Console.Read();”读取的是一个字符, 返回的是 int 值, 即为字符的 ASCLL 码, 按下任意按键即停止数据收集。

进行输入操作时, 还会涉及到对输入字符串的数据转化。数据转化有这么三种, “Parse”、“TryParse”和“Convert.To”

Parse 是接受字符串并解析字符串转化为指定的数据类型。格式为<Date-Type> Variable =<Date-type> Parse.(string\_date);  
//例子 2 将 date 转化为 int 类型

```
int date = int.Parse(Console.ReadLine());/*将输入的 string 的数据类型转换为 int 型*/
```

TryParse 会将尝试将输入的 string 数据类型转化成制定的数据类型, 如果转化成果, 并存成布尔变量 1, 若失败则转化为布尔变量 0。

```
<Datetype>.TryParse(string_date,out DateType_Variable);  
//例子 3, 如果可以将 date 可以转化为 int, 则输出 yes, 否则为 no
```

```
int date;  
if(int.TryParse(Console.ReadLine(),out date)==true)  
    {Console.WriteLine(“yes”);}  
else if(int.TryParse(Console.ReadLine(),out date)==false)  
    {Console.WriteLine(“no”);}
```

Convert.To 会接受字符串并尝试解析字符串成指定的数据类型并返回。格式为 Convert.To< DataType >(string\_data); 值得注意的是 <Datetype>格式应该为 int16, int32, int64, Single

```
//例子 4 将 date 转化为 in16 类型  
int date = Convert.ToInt16(Console.ReadLine());
```

## 二) 控制结构

首先 C#中有这么几个逻辑运算

==	等于	&&	且
!=	不等于		或
<	小于	!	非
<=	小于等于		
>	大于		
>=	大于等于		

这些运算符用于判断条件是真还是否。

控制结构有这么顺序结构，选择结构，循环结构。

### 1) 顺序结构

一条一条地执行语句。

### 2) 选择结构

根据某个条件的真假来觉得是否执行某个语句。涉及到 if, if else, switch-case 这几种。

对于 if 语句，基本格式为：

```
if(条件)
    {执行语句; }
```

if 语句进行嵌套：

```
if (条件 1)
    { 执行语句;
      if(条件 2)
      {.....}
    }
```

if 中多分支：

```
if (条件 1)
    执行语句;
else if (条件 2)
    执行语句;
...
...
...
else 执行语句;
```

```
//例 5
//例子选择PPT Week4 page 28
int mark = int.Parse(Console.ReadLine());
if ( mark < 0 || mark > 100 )
    Console.WriteLine("Please check... Mark should be in 0-100 ONLY");
//如果分数大于100或者小于0，重新输入
else if (mark >=79)
    Console.WriteLine("Grade: HD");//分数大于等于79，成绩为HD
else if (mark >=69||mark < 79)
    Console.WriteLine("Grade: D");//分数大于等于69小于79，成绩为D
else if (mark >=59||mark<69 )
    Console.WriteLine("Grade: C");//分数大于等于59小于69，成绩为C
else if (mark >=49||mark<59)
    Console.WriteLine("Grade: P");//分数大于等于49小于59，成绩为P
else
    Console.WriteLine("Grade: N");//分数小于49，成绩为N
```

对于 Switch-case 结构，会判断变量是否满足某一个条件，若满足某一条件，则执行该条语句；若所有条件都不满足，可以用 default 来判断。Switch-case 的结构为

```
switch(变量)
{
    Case<值>: 执行语句; break;
    ...
    ...
    ...
    default: 执行语句; break;
}
```

//例 6

//例子选自 PPT week4 page32

```
int DayOfWeek = int.Parse(Console.ReadLine());
switch(DayOfWeek)
{
    case 0: Console.WriteLine("Day 0 is Sunday."); break;
    // 若 DayOfWeek 值为 0，输出 Day 0 is Sunday.

    case 1: Console.WriteLine("Day 1 is Monday."); break;
    //若 DayOfWeek 值为 1，输出 Day 1 is Monday.

    case 2: Console.WriteLine("Day 2 is Tuesday."); break;
    //若 DayOfWeek 值为 2，输出 Day 2 is Tuesday.

    case 3: Console.WriteLine("Day 3 is Wednesday."); break;
    //若 DayOfWeek 值为 3，输出 Day 3 is Wednesday.

    case 4: Console.WriteLine("Day 4 is Thursday."); break;
    //若 DayOfWeek 值为 4，输出 Day 4 is Thursday.

    case 5: Console.WriteLine("Day 5 is Friday."); break;
    //若 DayOfWeek 值为 5，输出 Day 5 is Friday.

    case 6: Console.WriteLine("Day 6 is Saturday."); break;
    //若 DayOfWeek 值为 6，输出 Day 6 is Saturday.

    default: Console.WriteLine("That value is invalid."); break;
    //若 DayOfWeek 值为不在 0~6，输出 That value is invalid..
}
```

### 3) 循环结构

如果某些条件未满足,则将返回到改语句开头重复代码。C#中循环结构给出了如下几种,do-while、while、for、foreach。

do-while 循环。如果表达式为真,则执行该条语句,然后再次判断,只要表达式为真该语句就会继续重复。同时 do-while 循环中无论表达式是否为真都会将程序执行一次。Do-while 循环的基本格式如下:

```
do
{表达式;
...
...
}while(条件);
```

//例 7

//让 date 从 1 加到 100

```
int date=1, Counter = 1;
do
{
    Counter++;
    Date+=Counter;
}while(Counter<101);//date 从 1 加到 100
Console.WriteLine( "{0} ",date);
```

While 循环。如果表达式为真,则执行该条语句,然后再次判断,只要表达式为真该语句就会继续重复。与 do-while 不同的是,如果初始条件不满足,不会执行循环语句。While 循环格式为

```
while(条件)
{执行语句; }
```

//例 8

//让 date 从 1 加到 100

```
int date=1, Counter = 1;
while(Counter<101)
{
    Counter++;
    Date+=Counter;
} //date 从 1 加到 100
Console.WriteLine( "{0} ",date);
```

For 循环。For 的格式为

For(初始变量;条件;对初始变量改变)

{执行语句; }

For 循环可以轻松实现计数的功能。

```
//例 9
//date 从 1 加到 100
int date=1, Counter;
for(Counter = 1;Counter < 101;Counter++)
{
    Date+=Counter;
} //date 从 1 加到 100
Console.WriteLine( "{0} ",date);
```

Foreach 循环。该循环为 C#独有的循环语句，foreach 语句为数组或对象集合中的每个元素重复一组嵌入语句。foreach 语句用于遍历集合以获取所需信息，但不应用于更改集合的内容，以避免出现不可预测的副作用。

```
//例 10
//例子选择 PPT week5 page22
```

```
String sentence= "C# gives you programming power." ;
foreach (char item in sentence)
    Console.WriteLine(item);
```

//输出结果:

C

#

g

i

v

e

s

y

o

u

p

r

o

g

r

a

m

m

i  
n  
g  
  
p  
o  
w  
e  
r  
.

### 三) 数组:

数组就是一组相似数据的变量。数组储存的是多个相同类型的值，所有的变量都成为元素，且必须都是相同的数据类型。一维数组可以储存和处理一组数据。多维数组可以用于储存和处理多组数据。我们也可以对数组进行一些列的操作，比如求出数组的最大值，数组的平均数，给数组排序或者是删除其中的某一个元素等。

数组的结构如下:

<变量类型>[] 数组名 = new <变量类型>[数组大小]{各个元素};

其中我们要注意到，数组的大小为最大元素下标+1，元素下标从 0 开始。例如，若有 5 个元素，那么第一个元素下标 0，第五个元素下标为 4，那么数组大小为 4+1 = 5。数组大小也会根据 { } 中元素的个数自动填充。

//例 11

//一个有 5 个随机数的数组，并将所有的元素打印出来，并且相加，再求出其平均数。

```
int[] Array = new int[5];
    var Rd = new Random();

    int Counter;

    for(Counter=0; Counter<Array.Length;Counter++)
    {
        Array[Counter] = Rd.Next(1,100);
    }

    for(Counter=0;Counter<Array.Length ;Counter++ )
    {
        Console.WriteLine("{0}",Array[Counter]);
    }//将所有元素表达出来

    int tamp=0;
    for(Counter=0;Counter<Array.Length ;Counter++)
    {
        tamp += Array[Counter];
    }
```

```
Console.WriteLine("{0}", tamp/Array.Length); //所有元素相加并求平均  
均值
```

```
//输出:
```

```
85  
49  
96  
48  
42  
64
```

对于数组我们可以进行一些操作，例如删减其中某个元素，原理为将改元素替换为后面一个

```
For(int i=0;i<数组长度;i++)  
{  
    if(删除的某个元素)  
    {  
        For(int j=i;j<数组长度-1;j++)  
            {数组[j] = 数组[j+1]; //覆盖}  
        i--;  
    }  
}
```

#### 四) 方法:

方法 (method) 是一种用于实现可以由对象或者类执行的计算或操作的成员，将一些相关语句组织到一起，来执行一个任务的语句块。一般格式为

修饰符 返回值类型 方法名称 (参数)

```
{  
    方法体  
}
```

其中，修饰符有 public, private。Public 为公共类，其它类可以调用。Private 为私有类，私有类只能在本类中使用，其他类不可使用。返回值可以返回 string 值，或者是 int 值，布尔值。如果没有返回值则为返回的是 Void 类型。

方法中的参数为形参，在方法中可以指定它的地址 (out/ref/val/params)，形参不影响实参，方法被调用的时候创建，运行结束的时候摧毁，call-by-Value 是值的传递，而如 ref 是引用传递。

```
//例 12
```

```
//例子选自 PPT week7 page15
```



```

static void swap (ref string a, ref string b)
{ string temp = a; a = b; b = temp;
  Console.WriteLine("At the end of swap a: {0} b: {1}", a, b);
}

public static void Main()
{ string a = "Hello", b = "World"; swap(ref a, ref b);
  // a 存储包含 b 和 b 存储结果
Console.WriteLine("In Main after calling swap a: {0} b: {1}", a, b);
//转化("Helen Smith", "John Anderson");

```

类（class）是定义抽象数据类型的程序结构。类中又有字段，方法，属性等。字段是用来表述特征的，方式是表述对象的行为。类的默认修饰符是 internal（程序集），方法和字段默认修饰符是 private。其中修饰符有 internal（程序集），public（公共），private（私有），static（静态），protected（受保护的）。类的定义如下

修饰符 class 类名

```

{

    修饰符(private 等) 数据类型 (int 等)  变量参数;
    .....

    修饰符 类型 字段名
        {
            get {return 字段名; }
            set { 字段名 = value}
        }
    修饰符 class 类名 ( ) {};
    修饰符 class 类名 (值)
    {
        方法
    }
}

```

类在调用的时候会需要先创建实例化对象格式如下：

类名 实例化名 = new 类名 ( ) ;

```

//例 13
//例子选择 PPT week 7 page27-28
public class Name
{
    //property

```

```

    public string FirstName { get; set; }
    public string LastName { get; set; }
    public Name()
    {
        FirstName = "Unknown First Name";
        LastName = "Unknown Last Name";
    }
    public Name(string NewLastName, string NewFirstName)
    {
        FirstName = NewFirstName;
        LastName = NewLastName;
    }
}

public class Program
{ // no data or properties
    public static void GetData(string prompt, out string data)
    {
        do
        {
            Console.Write(prompt);
            data = Console.ReadLine();
        } while (data == "");
    }
    public static void Main()
    {
        // declare 2 string variables as test 3
        string LastName, FirstName;
        GetData("Enter First Name: ", out FirstName);
        GetData("Enter Last Name: ", out LastName);
        Console.WriteLine("After Get input data the name {0}, {1}",
LastName.ToUpper(), FirstName);
    }
}

```

//输出结果

*Enter First Name: James*

*Enter Last Name: May*

*After Get input data the name MAY, James*

## 五) 文件:

C#中对文件的操作分为三步骤, 打开文件 (如果不存在将会被创建), 从文件中读取数据或者将数据写入文件, 程序关闭文件。

### 1) 打开文件

C#若要打开文件，需要使用 I/O 流：

```
Using System.IO;
```

之后才能使用 StreamWriter，使用 StreamWriter 时要先对其进行声明：

```
StreamWriter 名称;
```

然后使用 CreatText 方法对文件进行操作：

```
名称.File.CreatText(“文件地址”)
```

文件地址为相对路径或者绝对路径。

### 2) 从文件中读取数据或者将数据写入文件

```
文件名.WriteLine(Data);
```

该方法的 Data 参数由常量或变量组成，其中包含要写入的数据，WriteLine 在数据末尾附加一个不可见的换行符。删除数据则为

### 3) 关闭文件

```
文件名.close();
```

C#中对文件的操作有 append, CreatText, StreamReader 等，使用方法同上类似，可以进行的操作为 append 附加文件，Creatext 创建一个文件，StreamReader 读取文件。

### 总结：

上述的五个章节，输入输出，控制结构，数组，方法，文件为本学期《编程概述(.NET)》学习到的主要知识点。分析，理解并且掌握这些基础语法十分重要。通过对这个五个章节的学习，我们掌握了 C#编程的基本方法，为以后学习 C#中例如异步，事件，订阅等功能，和未来从事 ASP.NET 开发，Unity 等打下基础。C#还要很多特性需要我们学习。同时 C#本身也只是一个开发工具，我们要用好这个工具。

## (二) 软件编程中相关领域技术发展动态分析

信息技术和计算机的发展给人们带来巨大的便利。并产生了互联网，大数据，云计算等数字革命。软件是计算机技术的载体，渗透到我们行业的各个方面。软件无所不在，航天飞船，大型服务器，制造业，金融行业抑或是我们平时使用的各种 APP 像 QQ，微信等应用。软件无所不在充斥着我们的生活，并将深刻改变我们的生活。

软件产业是国家重点发展的新兴产业。软件产业是数字经济中非常重要的一环，目前我国软件产业仍处于发展中，仍待提高。另外还要软件国产化，推动软件高质量发展等问题。工信部也召开《十四五软件与信息技术服务业发展规划》，推动软件产业发展，协同创新，提高关键技术等，并推动经济的增长[1]。同时，软件行业人才需求大，高端软件技术人才十分紧缺，我也在 2001 年设立国家性示范软件学院，培养一批软件方面的人才，一些城市也布局软件产业基地，推动软件行业发展。

信息技术推动软件行业进一步发展。4G 时代，网速的提高，延迟的降低，涌现了一批高质量，现象级别的软件，如抖音，美团，微信等等。互联网企业也迎来爆发式的增长。信息技术发展无疑会带动软件产业的发展的。如今 5G 时代的到来，低延迟，高带宽，万物互联的等，软件行业又会迎来发展。又随着智慧城市，智慧教育，智慧交通，智慧家居等等概念的兴起，软件会充斥我们的生活，一个“软件定义的世界”。

重视用户个人数据问题。今年 11 月《个人信息保护法》的实施，针对过度收集用户数据，大数据杀熟，互联网行业对用户信息处理等问题都做出了法律规范[2]。《个人信息保护法》的施行，对互联网和软件的健康发展非常重要。软件企业要重视起信息安全，保护用户隐私数据等问题，恪守职业道德，遵守相关的法律法规等。

### （三）软件编程中相关领域技术的发展趋势

人工智能和机器学习的应用。在编程开发中，人工智能的应用可以结合上下文自动填充，又或者是自动纠错，修复 bug 等，大大地提高了编程的效率。在软件服务上，人工智能的应用可以为用户个性化地推荐服务，如今大为流行的“短视频流”和社交软件等。或者金融服务，地图服务，旅游软件等，可以智能帮我们决策和生成个性化的旅游方案。人工智能是大趋势，相关研究的落地，

云计算的应用。云计算可以将大规模的计算分解各种小程序，通过网络，让多个服务器进行计算再返回结果[3]。云计算的优势在于，可以进行超大规模的计算，成千上万的大型计算机，大型服务器等，通过“云”进行算力整合，产生强大的算力。虚拟化，云计算并不是实体的服务，只要你的电脑，手机，平板等，可以接入互联网，就可以使用各种云计算带来的各种服务。价格较为低廉，云计算可以通过资源整合减少浪费，企业用户只需要缴纳少量租金就可以使用云服务了，如今大大小小的软件公司都是租用了其它公司的服务器。云计算在企业生产管理，如 iaas；网络云备份等。云计算成为了互联网第三次革命，云计算的应用也未来的大趋势。

虚拟现实和元宇宙。2016 年，被叫做 vr 元年，如今，元宇宙的概念又兴起。元宇宙概念是由增强现实，区块链，云计算，数字孪生等技术的整合。增强现实，VR 和 AR 等可以提供沉浸式的虚拟体验；区块链，提供虚拟社会下的经济流通；云计算，提供各种虚拟社会的服务等[4]。元宇宙要求的各种数字技术的整合，5G 技术，芯片技术，软件技术，人工智能等等的成熟，形成一套完整的体系，构建数字世界。其中软件是重点，元宇宙相关的软件开发也是一个趋势。

## （四）软件编程中相关前沿技术研究

人工智能。人工智能研究包含多个学科的交叉融合，计算机科学与工程，心理学，数学，生物学等。人工智能在软件产品已经有了广泛的应用。像苹果的 siri，小米的“小爱同学”等这些语音交互系统，或者是谷歌翻译，一些人工合成的以假乱真的声音等。又比如像自动驾驶，微博给我们推送文章，也都是人工智能的应用。又比如图形修复，智能决策等，谷歌开发者大会上展示的各种人工智能应用。人工智能分为三个层次。基础层，像芯片，技术框架等；技术层，有三大方面，机器学习，自然语言处理，计算机视觉等；应用层，也就是人工智能软件等，像我们所熟知的比如谷歌，百度等公司的软件。人工智能在是全球科技公司渴望占领的制高点，人工智能发展，计算机将会完成人类无法完成的工作

计算机视觉。计算机视觉是目前最热门的研究领域。计算机视觉的应用十分广泛，如今比较热门的自动驾驶，机器人，工业生产领域以及人机交互系统，计算机视觉技术都得到了应用。计算机视觉涉及到很多学科的交叉，涉及到人工智能，神经科学，计算机科学，图形处理等等。它所做的是让计算机可以感知环境，让它可以“看见”环境的变化。计算机可以“看见”环境后，经过一系列图形图像处理 and 人工智能的处理等，对所“看见”的环境做出改变[5]。

物联网。物联网就是让万物都连上互联网。人与物直接，人与人之间，可以进行信息的交流，实现智能化的管理和识别[6]。物联网让彼此独立的个体通过相同的协议连接到了一起，形成一个系统，提高效率。物联网的应用在工业生产，智慧建筑，智慧城市等。物联网是继计算机技术，互联网技术后信息技术产业的又一大浪潮。人工智能融合，形成 AIoT 和边缘计算，云计算，大数据等，又将成为物联网增长点。

## 参考文献

- [1] 工业与信息化部.《十四五软件与信息技术服务业发展规划》[R]. [2021-11-15]  
<https://wap.miit.gov.cn/>
- [2] 新华社客户端.《中华人民共和国个人信息保护法》[Z]. [2021-08-20]
- [3] 许子明，田杨锋. “云计算的发展历史及其应用”[J]. 信息记录材料 2018, 19(8):66-67
- [4] 新华社. “什么是元宇宙？为何要关注它？”[N]. [2021-11-20]

- [5] 郁梅, 蒋刚毅, 郁伯康. 智能交通系统中的计算机视觉技术应用[J]. 计算机工程与应用, 2001.
- [6] 刘陈, 景兴红, 董钢. “浅谈物联网的技术特点及其广泛应用” [J]. 科学咨询, 2011 (9): 86