

## 第16章 while文



while文も繰り返し処理を行います。C/C++のそれとほぼ同じ使い方をします。

while文は次のような形をとります。

```
while (条件式)
{
    ...
}
```

条件式がtrueの間...を反復実行します。条件式が最初からfalseの場合は...は一度も実行されません。また、...が1つの文であれば{}を省略できます。

while (true)とすれば無限ループになります。

C/C++のようにwhile(1)のような使い方は許されていません。

```
// while01.cs

using System;

class while01
{
    public static void Main()
    {
        bool bError = false;

        while (true)
        {
            Console.WriteLine("単位期間の利率(%) (0で終了) ---- ");
            string strR = Console.ReadLine();
            if (strR.CompareTo("0") == 0)
                break;

            for (int i = 0; i < strR.Length; i++)
            {
                if (!char.IsDigit(strR[i]) && strR[i] != '.')
                {
                    bError = true;
                    break;
                }
            }

            if (bError)
                break;

            double r = double.Parse(strR) / 100.0;

            Console.WriteLine("借入金額---- ");
            string strGan = Console.ReadLine();

            for (int i = 0; i < strGan.Length; i++)
            {
                if (!char.IsDigit(strGan[i]))
                {
                    bError = true;
                    break;
                }
            }
            if (bError)
                break;

            double a = double.Parse(strGan);

            Console.WriteLine("期間---- ");
            string strKikan = Console.ReadLine();

            for (int i = 0; i < strKikan.Length; i++)
            {
                if (!char.IsDigit(strKikan[i]))
                {
```

```

        bError = true;
        break;
    }
}

if (bError)
    break;
double n = double.Parse(strKikan);

double dTotal = a * Math.Pow((1.0 + r), n);
Console.WriteLine("元利合計は{0}円です。", dTotal);
}
}
}

```

これは、利率と借入金と、期間を入力することにより、元利合計を求めるプログラムです。利率は複利で計算されます。

### 元利合計 = 元金 \* (1 + 利率)の期間乗

で複利計算ができます。

Mainメソッドに、入るといきなりwhile文の永久ループです。

まず、利率を尋ねます。

ユーザーの入力した利率の文字列の参照をstrRに格納します。

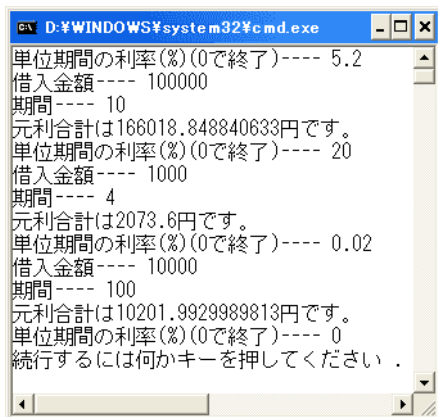
もしこの文字列が"0"ならば、break文でwhileループを抜けてプログラム終了です。CompareToメソッドについては、[第8章](#)に解説があります。

次にfor文で、strRの文字列を先頭から1文字ずつ検査します。10進数でもなくかつピリオドでもない場合は、bErrorをtrueにしてbreak文でforループを抜けます。ループを抜けた段階でbErrorがtrueなら、またbreak文に遭遇してwhile無限ループを抜けて、プログラム終了です。

どれにも、引っかからなかったら、double.Parse(strR) / 100.0で利率をdouble型の数値に変換します。

元金、期間についても同様です。

最後に、dTotal = a \* Math.Pow((1.0 + r), n)で元利合計を求めます。



次は、ユニコードでいわゆる全角の「あ」から「ん」にある文字を全部表示するプログラムを作ってみましょう。

// while02.cs

```
using System;
```

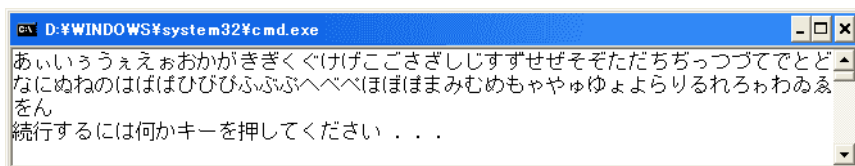
```

class while02
{
    public static void Main()
    {
        char a = 'あ';

        while (a <= 'ん')
        {
            Console.Write(a);
            a++;
        }
        Console.WriteLine("\n");
    }
}

```

実行結果は、次のようになります。



while文では、条件式が最初から偽の時は、一度も実行されない文ができてしまいます。最低でも1回は、実行してもらいたいときは、do..while文を使います。

```
do
{
    ...
} while (条件式);
```

条件式が真の間...が繰り返し実行されます。...は最低でも1回は実行されます。...がただ一つの文であれば、{}は省略可能です。また、最後にセミコロン(;)がついていることに注意してください。

次のプログラムでは1-9までの一桁の数字を入力してもらい(nとする)、 $1 + 2 + \dots + n$ の計算を行い、かつ計算式も表示するプログラムです。(単なるwhile文でも作れますが...)

```
// dowhile01.cs

using System;

class dowhile01
{
    public static void Main()
    {
        Console.Write("数字を入力してください (1-9) --- ");

        string strInput = Console.ReadLine();

        if (strInput.Length >= 2)
            return;

        if (!char.IsDigit(strInput[0]) || strInput[0] == '0')
            return;
        int n = strInput[0] - '0';

        string mystr = "";

        int i = 1, sum = 0;
        do
        {
            sum += i;
            mystr += string.Format("{0} + ", i);
            i++;
        } while (i <= n);

        char[] trimchar = new char[] { '+', ' ' };
        mystr = mystr.TrimEnd(trimchar);
        mystr += " = ";
        mystr += sum;
        Console.WriteLine(mystr);
    }
}
```

do..while文で

$1 + 2 + 3 + \dots + n +$

という文字列を作っていますが、最後の"+"は不要です。最後から文字を取り除くには stringクラスのTrimEndメソッドを使うのが便利です。

```
public string TrimEnd (
    params char[] trimChars
)
```

このオブジェクトの末尾からtrimChars配列の文字を全部取り除きます。

最後に"="と、sumを付け加えます。

ところで、

```
a += b;
```

は、 $a = a + b$ ;の意味です。同様に

```
a -= b;
```

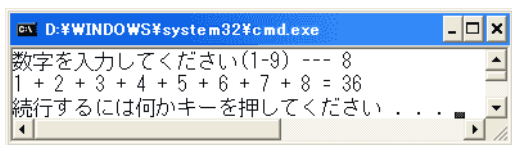
```
a *= b;
```

**a /= b;**

a = a - b, a = a \* b, a = a / bの意味になります。

これは、C/C++でもおなじみなので知っている人も多いでしょう。

実行結果は次のようになります。



---

[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 23/Aug/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。