Zeichenketten in C#

Der Datentyp char

Einzelne Zeichen werden in C# mit dem Datentyp `char` (sprich: Character) gespeichert. Werte von char-Variablen werden innerhalb des Programmtextes in Hochkommas eingeschlossen (z.B. 'a', 'x', '?', '2'). **Achtung:** '2' stellt das Zeichen 2 dar, nicht die Zahl 2.

Der Datentyp String

Im Gegensatz zu `char` ist ein String eine Aneinanderreihung von einzelnen Zeichen einer beliebigen Länge. Strings deklariert man als `String x;`
Werte von String-Variablen werden innerhalb des Programmtextes in Anführungszeichen eingeschlossen, z.B. "Meier", "Inf-12-Gk", "12&abc§7%?", "hallihallo" .

Auf einzelne Zeichen in einem String greift man mit eckigen Klammern zu. In den Klammern steht die Position des gewünschten Zeichens, die *von 0 gezählt* wird. `s[2]` liefert für einen String s, der das Wort "Wurm" enthält, das Zeichen ‚r'.

Zeichenketten haben eine große Anzahl von Methoden und Eigenschaften. Einige davon sind:

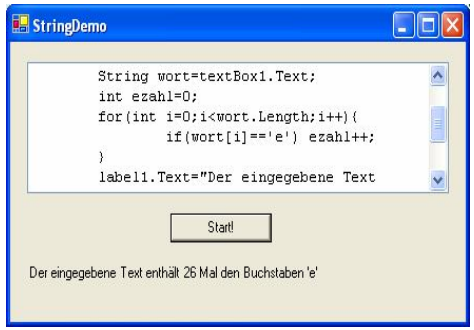
`n=s.Length;`	n: int s: String	Mit Hilfe der Eigenschaft Length wird die Länge des Strings s bestimmt, d.h. die Anzahl der Zeichen, aus denen s besteht. Beispielsweise erhält man für s= "Turm"; durch s.Length den Wert 4. *Nach Length steht keine Klammer!*
`z=s.Remove(p, n);`	p, n: int s, z: String	Beim String s werden ab der Position p n Zeichen gelöscht (höchstens aber bis zum Ende des Strings) und als Ergebnis zurückgeliefert. z.B.: `s= 'WACHTURM'; s=s.Remove(1, 4);` ergibt s ='WURM'
`z=s.Insert(p, s1);`	z, s, s1: String; p: Integer;	Der String s1 wird in s ab der Position p eingesetzt und ein neuer String erzeugt, z. B. `s="NORDWEST"; s=s.Insert(4, "NORD");` ergibt `s ='NORDNORDWEST'`
`r = s.Substring(p,n);`	r, s: String p, n: int	Aus dem String s werden ab der Position p n Zeichen (höchstens jedoch bis zum Ende des Strings) in die Zeichenkette r kopiert. Beispielsweise liefert `s="ABCDE";` `r=s.Substring(1,3);` in r die Zeichenkette "BCD".
`r = s.Substring(p);`	r, s: String p: int	Aus dem String s werden ab der Position p alle Zeichen bis zum Ende von s in die Zeichenkette r kopiert.
`p=z.IndexOf(q);`	p: int q, z: String	Es wird untersucht an welcher Stelle der String q im Zielstring z erstmals vorkommt. Das Ergebnis ist die Nummer dieser Stelle. Kommt q in z nicht vor, ist p = -1. z. B. `z="Hallo Welt"; z.IndexOf(" ")` ergibt 5
`p=z.IndexOf(q,r);`	q, r: int q, z: String	Im String z wird jetzt ab der Position r nach der Zeichenkette q gesucht. p liefert die Position.
`z = s.Replace(alt, neu);`	z, s: String alt, neu: String oder char	Durch die Methode Replace wird in der Zeichenkette s jedes Vorkommen von alt durch neu ersetzt und in einem neuen String gespeichert.

Eine Zahl wandelt man in einen String mit Hilfe der Methode ToString() der Zahl um, z. B. für `int x=3;` erhält man die dazu passende Zeichenkette durch `x.ToString();`
Einen String wandelt man in eine Zahl durch die Methode `Parse` der Zahl um. Diese wurde aber schon früher besprochen.

Wichtig:
Bei allen Methoden, die den Inhalt einer Zeichenkette verändern, wird die geänderte Zeichenkette zurückgeliefert. Die ursprüngliche Zeichenkette bleibt unverändert.

Beispiel:

```
void Button1Click(object sender,
                  System.EventArgs e)
{
      String wort=textBox1.Text;
      int ezahl=0;
      for(int i=0;i<wort.Length;i++){
            if(wort[i]=='e') ezahl++;
      }
      label1.Text=
        "Der eingegebene Text enthält "
        +ezahl.ToString()+
        " Mal den Buchstaben 'e'";
}
```



Dieses kleine Beispielprogramm durchsucht eine eingegebene Zeichenkette Stelle für Stelle auf das Vorkommen des Zeichens "e",

Zwei Zeichenketten können auch mit < oder > verglichen werden. Dabei werden die Strings von links nach rechts zeichenweise verglichen. Beim ersten Auftreten von ungleichen Zeichen entscheidet die Lage (Nummer) im Unicode-Zeichensatz (s.sp.) über kleiner oder größer. Strings kann man auch mit der Methode Compare vergleichen.

Beispiele: Es gilt: "balkon" > "balken" und "Georg" < "Hans"
Aber es gilt nicht: "Pascal" == "pascal"

Aufgaben:

1. Ein Text von max. 35 Zeichen soll eingelesen werden und g e s p e r r t ausgegeben werden. Nach jedem Zeichen ist also ein Leerzeichen auszugeben.
2. Ein C#-Programm soll bei Eingabe eines Wortes entscheiden, ob es sich dabei um ein so genanntes *Palindrom* handelt. Recherchiere gegebenenfalls im Internet.
3. Ein eingegebenes Wort ist von hinten nach vorne gelesen in einer zweiten String-Variablen zu übergeben. Danach sind diese beiden Wörter aneinander zu hängen und das so entstandene neue Wort auszugeben.
4. In einem Text sollen die Umlaute ä, ö, ü, Ä, Ö, Ü durch ae, oe, ue, Ae, Oe, Ue und ß durch ss ausgedrückt werden. Das Programm soll zwei Lösungen enthalten. Eine mit Hilfe der Methoden von Strings, eine ohne diese Methoden.
5. Nach Eingabe eines Textes soll der Text Wort für Wort untereinander in einer Listbox ausgedruckt werden. Das heißt in jeder Zeile soll nur ein Wort stehen.
Verwende für das Zerlegen keine fertigen Methoden der Klasse String.
6. Setze die Umwandlungen des Lieds „Drei Chinesen mit dem Kontrabass" in C# um.