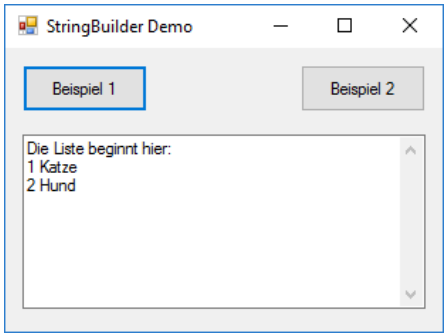
Zeichenketten und die StringBuilder-Klasse in C#

Wenn man Zeichenketten (Strings) in C# mit + zusammenbaut, ist dies nicht besonders performant, da für jede Addition eine neue Zeichenkette gebaut wird, die meistens nur einmal benötigt wird. Für die neue Zeichenkette wird der Inhalt des alten Strings kopiert, weil Strings bzw. String-Objekte in C# nämlich unveränderlich sind. Bei recht aufwendigen Anwendungen wird damit viel Speicher und Zeit für die Verwaltung der Objekte verschwendet.
Als Alternative bietet sich die Klasse **StringBuilder** an, die das Verändern des Inhalts zulässt. Schauen wir dazu ein erstes kleines Beispiel an:

```
using System.Text;  // um StringBuilder nutzen zu können

[…]

void Button1Click(object sender, EventArgs e)
{
    // Ein Objekt der StringBuilder Klasse erzeugen
    StringBuilder builder = new StringBuilder();

    // wichtige Methoden
    builder.Append("Die Liste beginnt hier:");
    builder.AppendLine();
    builder.Append("1 Katze").AppendLine();
    builder.Append("2 Hund").AppendLine();

    // Inhalt von builder in Zeichenkette
    string inhalt = builder.ToString();

    // Ausgabe
    textBox1.Text = inhalt;
}
```

Ganz wichtig ist die Zeile mit „using System.Text“, die man ganz oben in der Quelltextdatei hinzufügen muss. Ansonsten kann man StringBuilder nicht nutzen.
Folgende Fehlermeldung erscheint ohne die using-Zeile: *„Der Typ- oder Namespacename 'StringBuilder' konnte nicht gefunden werden. (Fehlt eine Using-Direktive oder ein Assemblyverweis?)“*



StringBuilder ist die erste Klasse, bei der man eine zusätzliche Assembly bzw. einen Namespace selbst einbinden muss. Bei den InfoKurs-Tools wurde dies schon in der Vorlage gemacht. Die komplette C#-Bibliothek mit ihren Klassen und Komponenten befindet sich in vielen verschiedenen Assembly-Dateien.

Mit `StringBuilder builder` wird eine neue Variable für ein Objekt namens builder der Klasse StringBuilder deklariert. Mit `new StringBuilder();` wird dieses neue Objekt erzeugt.

Mit `Append` kann man Elemente an die Zeichenkette im StringBuilder anhängen. Hier ist es eine Zeichenkette, aber auch Zahlen und andere Objekte sind möglich.

Mit `AppendLine` wird ein Zeilenumbruch angehängt (unter Windows CR+LF).

Da Append und AppendLine immer das aktuelle StringBuilder-Objekt zurückliefern, kann man die Methoden-Aufrufe einfach aneinanderhängen. Dieser Stil ist im Moment recht modern und wird „fluent coding“ genannt.

Mittels der ToString-Methode kommt man an die Zeichenkette heran, die im Objekt builder erzeugt wurde.

Die StringBuilder-Klasse hat auch Eigenschaften und Methoden zum Bestimmen der Länge der erzeugten Zeichenkette (Eigenschaft `Length`), zum Zugriff auf einzelne Zeichen (wie bei Strings: [ ]), zum Löschen, Einsetzen und Ersetzen einzelner Zeichen bzw. Teilstrings:

```
void Button2Click(object sender, EventArgs e)
{
    // neues Objekt mit Start-Text erzeugen
    StringBuilder sb = new StringBuilder(
      "Das ist der Beginn eines Satzes, ");

    // Ersetzen von Teilstrings (alle werden ersetzt!)
    sb.Replace(", ",".");
    // zuerst alten Wert, dann neuen Wert angeben

    // Teilstring finden und löschen
    string teilstring="der Beginn ";
    sb.Remove(sb.ToString().IndexOf(teilstring),11);
    // zuerst Startposition, dann Länge des zu löschenden
    // Bereichs angeben

    // neuen Teilstring einfügen
    sb.Insert(8,"ein neuer Anfang ");
    // zuerst Startposition, dann neuen Teilstring angeben

    // Länge auch ausgeben
    sb.AppendLine().Append("Länge :"+sb.Length);

    // Ausgabe
    textBox1.Text = sb.ToString();
}
```

Mehr Informationen zur StringBuilder-Klasse gibt es direkt bei Microsoft unter
https://msdn.microsoft.com/de-de/library/system.text.stringbuilder(v=vs.110).aspx