

Łańcuchy znaków - teksty

Podstawowym typem danych, który pozwala na przechowanie łańcuchów znaków jest typ „string”.

Przykład definicji zmiennej tekstowej:

```
string napis = „Programowanie”;
```

Łańcuchy znaków

W języku C# łańcuchy znaków traktowane są jako obiekty, na których możemy wykonywać przeróżne operacje związane z kontekstem łańcuchów znaków sortowanie, przeszukiwanie , itd..

Każdy obiekt typu *string* to niezmienna sekwencja znaków w formacie Unicode.

Niezmienność oznacza, że metody pozornie zmieniające zawartość zwracają jego zmodyfikowaną kopię.

Oryginalny łańcuch znaków pozostaje w pamięci do czasu, kiedy nie usunie go mechanizm odzyskiwania pamięci.

Tworzenie łańcuchów znaków

- ⦿ Najczęstszą metoda tworzenia znaków jest przypisanie do zmiennej typu *string* znaków umieszczonych w cudzysłowach. Taki zbiór znaków to literał znakowy.
- ⦿ Tekst w cudzysłowach może zawierać stałe symboliczne np. `\n`, `\t`.
- ⦿ `string s1 = "C:\\Programy\\C#";` //literał znakowy
- ⦿ `string s2 = @"C:\\Programy\\C#";` //literał dosłowny
- ⦿ Powyższe deklaracje są równoważne.

Tworzenie łańcuchów znaków

Metoda ToString()

- © Innym sposobem tworzenia łańcuchów znaków jest wywołanie do obiektu metody ToString() i przypisanie wyniku do zmiennej typu *string*.

```
int liczba = 225;
```

```
string tekst = liczba.ToString();
```

- © W przykładzie metoda ToString() powoduje zapisanie liczby całkowitej jako łańcuch znaków.

Manipulowanie łańcuchami znaków

Klasa *string* oferuje wiele użytecznych metod narzędziowych:

◎**Compare()** przeciążona publiczna metoda statyczna, która porównuje dwa łańcuchy znaków

◎**Concat()** przeciążona publiczna metoda statyczna, która tworzy nowy łańcuch znaków z jednego lub kilku innych łańcuchów.

◎**Join()** przeciążona publiczna metoda statyczna, która łączy elementy tablicy w łańcuch znaków.

◎**CompareTo()** porównuje łańcuchy znaków.

◎**Insert()** wstawia określone znaki i zwraca nowy łańcuch znaków.

Manipulowanie łańcuchami znaków

- ◎ **LastIndexOf()** zwraca indeks ostatniego wystąpienia danego znaku lub łańcucha.
- ◎ **Remove()** usuwa określoną liczbę znaków.
- ◎ **Split()** zwraca podłańcuch ograniczony znakami określonymi w tablicy.
- ◎ **Substring()** pobiera podłańcuch.
- ◎ **ToCharArray()** kopiuje znaki z łańcucha do tablicy znaków.
- ◎ **ToLower()** zwraca kopię łańcucha, w której wszystkie litery są małe.
- ◎ **ToUpper()** zwraca kopię łańcucha, w której wszystkie litery są wielkie.
- ◎ **Trim()** usuwa z początku i końca łańcucha wszystkie wystąpienia określonych znaków.

Metody

- ◎ **Compare ()** publiczna metoda statyczna, która służy do porównywania dwóch łańcuchów znaków. Zwraca ona liczbę ujemną w przypadku, gdy pierwszy łańcuch jest mniejszy od drugiego, zaś dodatnią – gdy mamy sytuację odwrotną. Jeśli oba porównywane ze sobą łańcuchy mają tę samą długość, wówczas *Compare()* zwraca liczbę 0.
- ◎ **EndsWith()** metoda ta określa, czy dany łańcuch znaków pasuje do końcowego fragmentu tego łańcucha (jeśli tak, to metoda zwraca *True*)
- ◎ **Właściwości:**
- ◎ **IsNullOrEmpty** sprawdza, czy dany łańcuch jest pusty
- ◎ **Length** określa liczbę znaków danego łańcucha

Metoda Compare()

```
string s1 = "nowy rok";  
string s2 = "NOWY ROK";  
  
int wynik; //przechowuje wynik porównania  
  
wynik = string.Compare(s1,s2); //uwzględnia wielkość liter  
  
Console.WriteLine("Metoda compare() s1:{0}, s2:{1}, wynik:{2}\n", s1, s2, wynik);  
  
//Przeciążone porównywanie, przyjmuje wartość logiczną  
  
//określającą, czy należy zignorować wielkość liter  
  
wynik = string.Compare(s1, s2, true);  
  
Console.WriteLine("Nie uwzględnia wielkości liter\n");  
  
Console.WriteLine("Metoda compare() s1:{0}, s2:{1}, wynik:{2}\n", s1, s2, wynik);
```


Metoda Concat()

//Metoda łączenia łańcuchów

```
string s = string.Concat(s1, s2);
```

```
Console.WriteLine(
```

```
"s {0} utworzony z połączenia s1:{1}, s2:{2}\n",
```

```
s, s1, s2);
```

// w tym celu można użyć przeciążonego operatora +

Equals()

//Porównywanie

```
Console.WriteLine(  
"Czy s1: {0} jest równy s2: {1}?: {2}\n",  
s1, s2, s1.Equals(s2)  
);
```

```
Console.WriteLine(  
"Czy s1: {0} jest równy s2: {1}?: {2}\n",  
s1, s2, string.Equals(s1,s2)  
);
```

```
Console.WriteLine(  
"Czy s1: {0} jest równy s2: {1}?: {2}\n",  
s1, s2, s1==s2  
);
```

Właściwości długość i indeks

//Właściwości długość i indeks

```
Console.WriteLine(  
    "Długość łańcucha s1:{0} wynosi {1}\n",  
    s1, s1.Length);
```

```
Console.WriteLine(  
    "Szósty znak łańcucha s1:{0} to {1}\n",  
    s1, s[5]);
```

IndexOf()

//Indeks podłańcucha

Console.WriteLine(

"\nPierwsze wystąpienie słowa \"rok\" w
s1: \"{0}\" to {1},

s1, **s1.IndexOf("rok")**);

Insert()

//Wstawianie łańcucha znaków

```
Console.WriteLine
```

```
(
```

```
"Nowy tekst: {0}",
```

```
s1.Insert(s1.IndexOf("rok"), "lepszzy ")
```

```
);
```

Pomocna strona:

http://4programmers.net/C_sharp/Wprowadzenie/Rozdzia%C5%82_9

Zadania

Zadanie 1

Napisz program, który utworzy tablicę dwuwymiarową znakową o liczbie elementów podanej przez użytkownika i wpisze do niej kilka osób (imię i nazwisko). Następnie pozwoli odszukać imię osoby po jej nazwisku.

Zadanie 2

Napisz program, który wyświetli wprowadzony przez użytkownika łańcuch znaków tak, by każdy znak znajdował się w osobnym wierszu.

Zadanie 3

Napisz program, który podany przez użytkownika ciąg znaków zapamięta w nowej zmiennej `wspak`.

Zadanie 4

Napisz program, który z podanego ciągu znaków usunie spacje wielokrotne.

Zadanie 5

Napisz program, który policzy ile jest słów w danym tekście.

Zadanie 6

Napisz program, który pomiędzy imieniem a nazwiskiem wstawi drugie imię. Następnie z powstałego tekstu usunie pierwsze imię. Potem utworzy z drugiego imienia i nazwiska inicjały.

Zadanie 7

Napisz program, który z podanego symbolu materiału wyciągnie ciąg znaków stanowiących datę w zapisie dd-mm-rrrr.

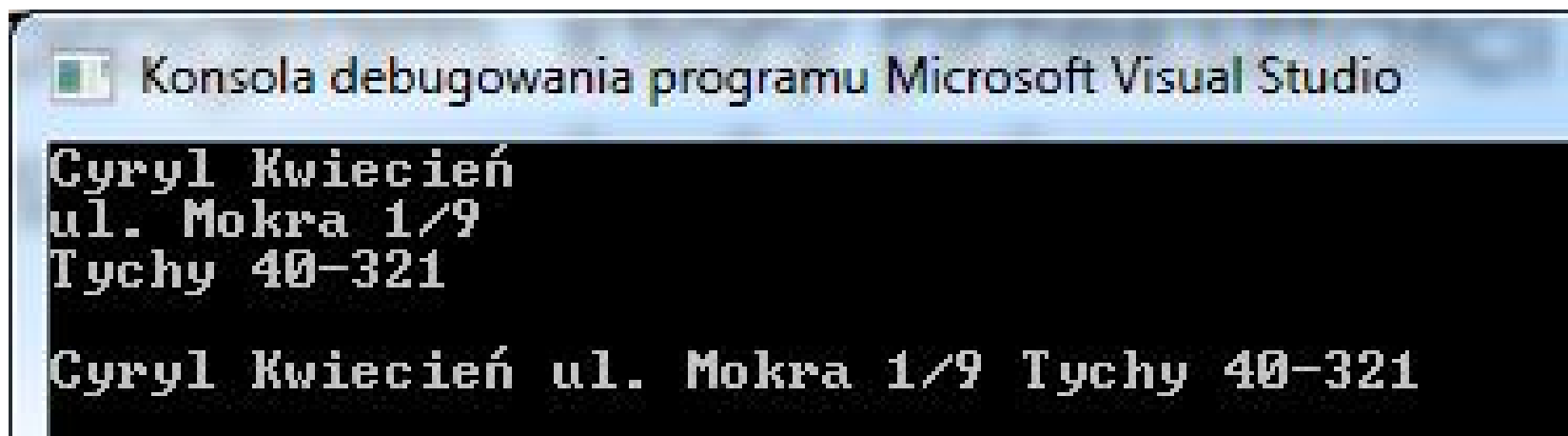
AA-1997-12-12-klej

ZZZZZ-2004-11-23-farba

BBBBBBB-2021-01-12-tapeta

Zadanie 8

Napisz program, który przekształci zapis trzywierszowy na jednowierszowy.

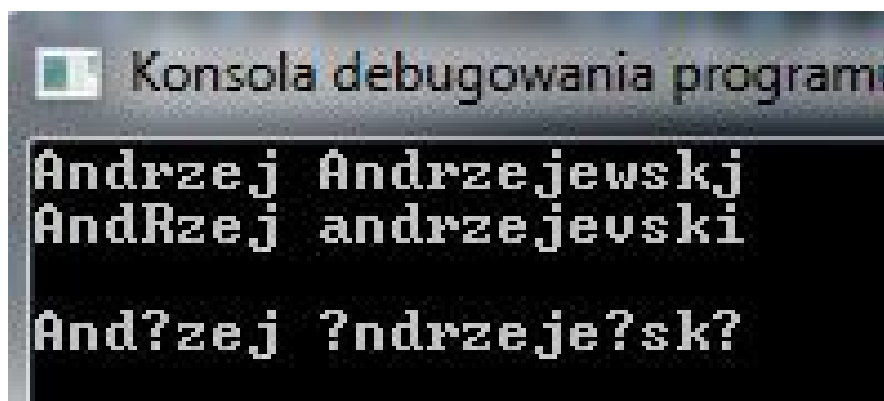


The screenshot shows a window titled "Konsola debugowania programu Microsoft Visual Studio". The console output consists of two lines. The first line contains the text "Cyryl Kwiecień", "ul. Mokra 1/9", and "Tychy 40-321" on three separate lines. The second line contains the same text concatenated into a single line: "Cyryl Kwiecień ul. Mokra 1/9 Tychy 40-321".

```
Konsola debugowania programu Microsoft Visual Studio  
Cyryl Kwiecień  
ul. Mokra 1/9  
Tychy 40-321  
  
Cyryl Kwiecień ul. Mokra 1/9 Tychy 40-321
```

Zadanie 9

Napisz program, który porówna dwa teksty o tej samej liczbie znaków. Jeśli wystąpi różnica to dany znak zostanie zastąpiony pytajnikiem.



```
Konsola debugowania programu  
Andrzej Andrzejewskj  
AndRzej andrzejewski  
And?zej ?ndrzej?sk?
```