**Урок 29: Тип низ – част 1**

**1. Тип на данните:**

**а) същност** - начинът, по който се тълкува записаното в едно поле на паметта

**б) характеристики**

* уникално име на типа
* отделената памет
* множество от допустими стойности
* допустими операции

В езика C#, както и в другите езици за програмиране, за обозначаване на типовете данни се използват ключови думи.

**в) константи и променливи**

* същност
* **константа** - всяка конкретна стойност на типа
* **променлива** – използва се за съхраняване на данните, необходими за работата на една програма
* **име (иднтификатор)**
* **деклариране на променлива** - в оперативната памет се определя поле с размера, необходим за съхраняване на стойности от посочения тип

*<име на тип> <име на променлива>;*

**Пример**и: int x;

 double pi;

 string name;

 int x, y, z;

* **инициализиране на променлива** – задаване на начална стойност на променлива. Това може да стане или по време на декларирането или в оператор за присвояване, преди променливата да бъде използвана.

Инициализирането на променлива може да стане в оператора за деклариране на променливата

<име на тип> <име\_на\_променлива> = <константа> ;

**Пример**: int x=0, y=-1, z=3;

или по-късно, но непременно преди стойността на променливата да бъде използвана за пръв път, с най-простия вид оператор за присвояване на стойност:

<име\_на\_променлива> = <константа>; или <име на променлива>=<променлива>.

**Пример:** int x, y, z;

x=0;

 y=x; z=3;

**2. Типовете char и string**

В езика C# знаците се представят с поредния си номер в двубайтовата таблица Unicode, в която има 216 = 65536 знака. Стандартът Unicode е създаден в края на 80-те и началото на 90-те години на миналия век, с цел съхраняването на текстови данни на различни езици. Да напомним, че пред-шественикът му, таблицата ASCII, позволява записването на едва 128 (в 7-битова версия) или 256 знака (в 8-битова версия). За съжаление, това не удовлетворява съвременните изисквания, особено публикуването в интернет, тъй като в 128 позиции могат да се поберат само цифрите, малките и главни латински букви и някои специални знаци, а в 255 – и кирилицата, но това е всичко. Ако се наложи работа с текст на кирилица, латиница и някаква трета азбука, например, 256 знака са крайно недостатъчни. Ето защо все по-често използваме 16-битовата кодова таблица Unicode. Низовете във всеки език за програмиране са последователности от знаци.

* *Константите от типа char* се записват, като съответният знак, когато го има на клавиатурата, се постави между два апострофа. Например 'А', '8' или '@'. В противен случай се използва кодът на знака в таблицата Unicode, предшестван от указание за преобразуване char, например (char)1040.
* *Константните низове* са последователност от знаци от таблицата Unicode, поставени в кавички. Например, "hello", "Hello", "My name is: ", "=" и т.н. Малките и главните букви имат различни кодове, т.е. са различими (низът “hello” е различен от низа "Hello").
* *Декларирането на променливи от тип char и string* става по обичайния начин. Например, char x; string name; char a, b, c; При деклариране на променлива х от типа char в паметта ще бъде заделено поле с размер 2 байта, тъй като това е размерът за този тип. Количеството памет, заделено за променлива от типа string, не зависи от дължината на съхранения в нея низ, защото тя съдържа не самия низ, а адреса на полето от паметта, където той е разположен.
* *Инициализирането на променливи от тип char и string* може да стане с оператор за деклариране:

*char <променлива от типchar> = <константа>;*

*string <променлива от тип string> = <константа>;*

или

*char<променлива от типchar> = <променлива от типchar>;*

*string<променлива от тип string> = <променлива от типstring>;*

Например, char x = '\*', y = 'a';

 char z = x;

 string s = "Здравей";

 string s = t;

* *Въвеждане* - когато създаваме конзолно приложение, за да дадем на потребителя да въведе желан от него низ, използваме метода Console.ReadLine():

*string s = Console.ReadLine();*

В приложение с графичен интерфейс за инциализиращ низ може да се вземе съдържанието на текстово поле (textBox1) или надписа на етикет (label1) чрез свойството Text:

*string s1 = textBox1.Text;*

*string s2 = label1.Text;*

* *Извеждането* на низ в конзолно приложение става с метода WriteLine(), в скобите на който се изписва име на променлива или константа от тип string:

*string s = "Здравей";*

*Console.WriteLine(s);*

*Console.WriteLine("Въведи име:");*

В графично приложение за извеждане на низ най-често се използват интерфейсните компоненти текстово поле и етикет и свойството им Text :

*label1.Text = "My name is:";*

*textBox1.Text = "Ivan";*

* *сливане*

Естествена операция с низове е **сливането** (конкатенацията) на два низа. Конкатенацията на низовете α и β е низът αβ. В езика C# конкатенацията на низове е със знак +. Тази операция не е симетрична – има значение кой от операндите е отляво и кой отдясно на знака +. В примера:

*string s1 = "Иван", s2 = "Борис";*

*string s3 = s1 + s2;*

*string s4 = s2 + s1;*

низовете s3 иs4 получават съответно стойностите "ИванБорис" и "БорисИван", които очевидно са различни. Могат да се сливат и повече от два низа, като между всеки два от тях се слага знакът +. Например, ако искаме да обединим съдържанието на две текстови полета в трето, но между тях трябва да има за разделител интервал, то трябва да добавим и низ с интервал " " в конкатенацията:

*textBox3.Text = textBox1.Text + " " + textBox2.Text;*

В изрази с низове могат да участват и числови стойности, тъй като всяка числова стойност може да се представи с десетични цифри, точка и знаците плюс и минус. Когато е включено в израз с низове, всяко число се преобразува в низа, който е десетичното му представяне:

*string s = "Иван е на " + 14 + " години";*

Константната стойност на низа s е "Иван е на 14 години";

**Дом. стр.57-58, зад. 1, 2, 3 след урока**