

# Други типове данни

*Знаков, низов и обектен тип данни,  
преобразуване между типовете*

Увод в  
програмирането



# Съдържание

1. Знаков тип
2. Низове
3. Преобразуване на типове



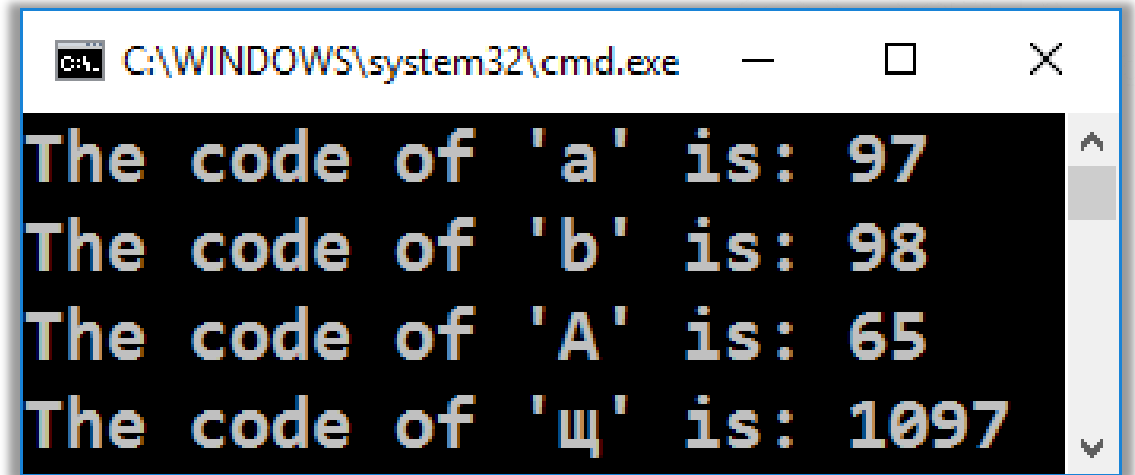
# Знак

- Типът данни знак в C#
  - Съдържа информация за един символ
  - Декларира се с **char** ключовата дума
  - Всеки символ съответства на числов код
  - Стойността по подразбиране е **\0**
  - Заема 16 бита в паметта (от **U+0000** до **U+FFFF**)
  - Съдържа един Уникод знак (или част от знак)



# Знаци и кодове

- Всеки знак има уникална цяла Уникод стойност (**int**):



```
C:\WINDOWS\system32\cmd.exe
The code of 'a' is: 97
The code of 'b' is: 98
The code of 'A' is: 65
The code of 'щ' is: 1097
```

```
char ch = 'a';
Console.WriteLine("The code of '{0}' is: {1}", ch, (int) ch);
ch = 'b';
Console.WriteLine("The code of '{0}' is: {1}", ch, (int) ch);
ch = 'A';
Console.WriteLine("The code of '{0}' is: {1}", ch, (int) ch);
ch = 'щ'; // кирилската буква „щ“
Console.WriteLine("The code of '{0}' is: {1}", ch, (int) ch);
```

# Знакови литерали

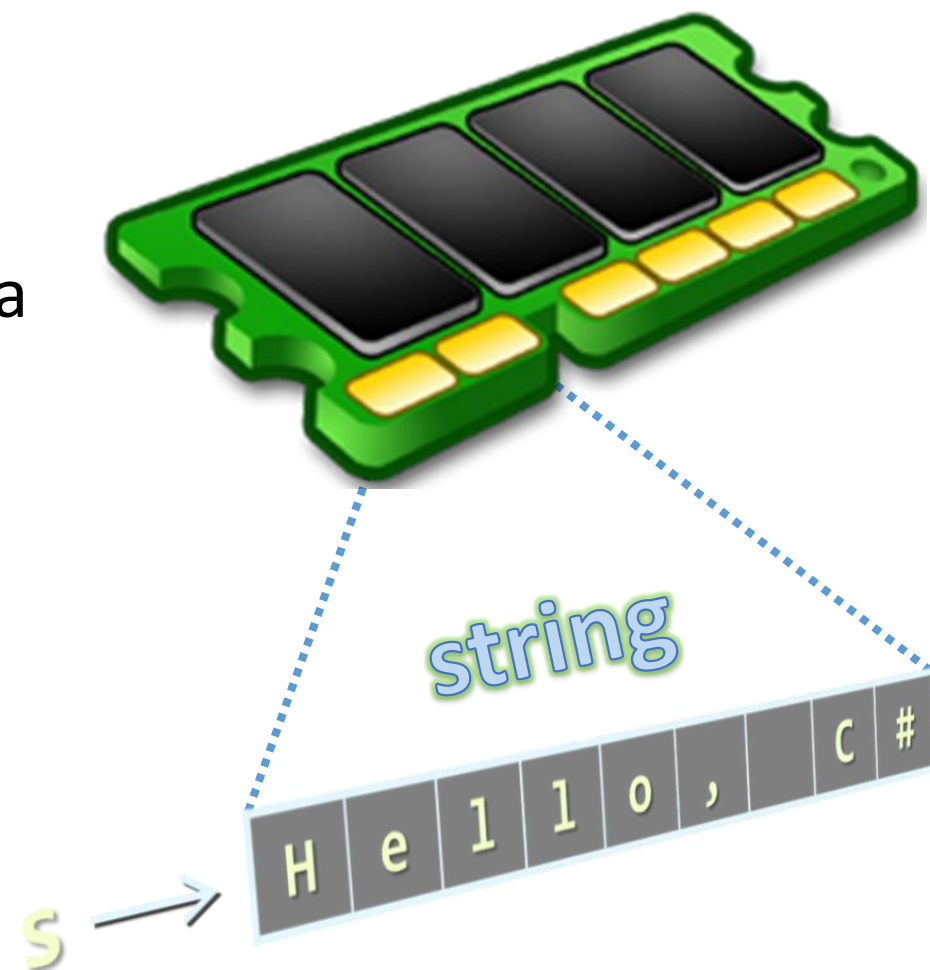
- Знаковите литерали се ограждат с апостроф – например 'A'
- Екраниращите последователности се ползват, когато:
  - Съдържат специален знак като ' , " или \n (нов ред)
  - Съдържат системни знаци (като [TAB] - знакът \t)
- Често срещани екраниращи последователности:
  - \' → апостроф      \" → двойна кавичка
  - \\ → наклонена черта      \n → нов ред
  - \uXXXX → за отбелязване на кой да е Уникод символ

# Знакови литерали – примери

```
char symbol = 'a'; // Обикновен знак
symbol = '\u006F'; // Уникод знак
                        // в 16-ичен формат (буква 'o')
symbol = '\u8449'; // 葉 (Листо в Традиционен китайски)
symbol = '\''; // Присвояване на апостроф
symbol = '\\'; // Присвояване на наклонена
symbol = '\n'; // Присвояване на знак за нов ред
symbol = '\t'; // Присвояване на знак TAB
symbol = "a"; // Неправилно: използвайте апострофи!
```

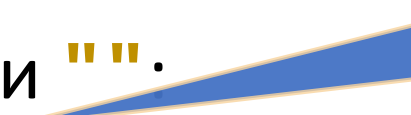
# Низове

- **Низовете** в C#
  - Представят поредица от знаци
  - Задават се чрез **string** ключова дума
  - Имат стойност по подразбиране: **null** (празна стойност)
- Низовете се обграждат с кавички:  
**string s = "Hello, C#";**
- Низовете могат да се слепват
  - Чрез оператор **+**



# Дословни (verbatim) и съставни (interpolated) низове

- Низовете са оградени от кавички "".


Наклонената черта \  
се екранира от \\  


```
string file = "C:\\Windows\\win.ini";
```

- Низовете могат да са **дословни** (без екраниране):

```
string file = @"C:\Windows\win.ini";
```

- Съставните** низове могат да съдържат стойности на променливи по шаблон:

Наклонената черта \  
не се екранира  


```
string firstName = "Svetlin";  
string lastName = "Nakov";  
string fullName = $"{firstName} {lastName}";
```



# Кажи „здрасти“ – Примери

- Комбиниране имената на човек, за да получите пълното име:

```
string firstName = "Ivan";  
string lastName = "Ivanov";  
Console.WriteLine(@"Hello, ""{0}""!", firstName);  
  
string fullName = $"{firstName} {lastName}";  
Console.WriteLine("Your full name is {0}.", fullName);
```

- Можем да слепим низовете с оператор +:

```
int age = 21;  
Console.WriteLine("Hello, I am " + age + " years old");
```

## Задача: Поздрав по име и възраст

Напишете програма, която въвежда малкото име, фамилията и възрастта и извежда "Hello, <first name> <last name>. You are <age> years old."

```
string firstName = Console.ReadLine();  
string lastName = Console.ReadLine();  
string ageStr = Console.ReadLine();  
int age = int.Parse(ageStr); // от string → int  
Console.WriteLine($"Hello, {firstName} {lastName}.  
                        \r\nYou are {age} years old.");
```

# Преобразуване на типове

- Променливите съдържат **стойности** от даден **тип**
- **Стойността** може да се промени (**преобразува**) към друг тип
  - **Скрито** преобразуване на тип (без загуби): променлива от по-"голям" тип (пр. **double**) взема по-"малка" стойност (пр. **float**)

```
float heightInMeters = 1.74f;  
double maxHeight = heightInMeters; //Скрито преобразуване
```

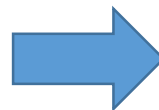
- **Явно** преобразуване (със загуба) – може да загубим точност:

```
double size = 3.14;  
int intSize = (int) size; //Явно преобразуване → 3
```

## Задача: Асансьор

- Изчислете колко курса са нужни, за да се качат **n** човека с асансьор с **капацитет от p** човека

Брой хора = 17  
Капацитет = 3



6 курса

Как? 5 курса \* 3 човека  
+ 1 курс \* 2 човека

- Просто решение:

```
int n = int.Parse(Console.ReadLine());  
int p = int.Parse(Console.ReadLine());  
int courses = (int) Math.Ceiling((double)n / p);  
Console.WriteLine(courses);
```

# Преобразуване с Convert

- **Convert** позволява да се извършват преобразувания на данни:
  - **Convert.ToInt32**(данни, основа) – от низ, съдържащ число в позиционна бройна система с подадена основа към число от тип int

```
int nums = Convert.ToInt32(Console.ReadLine(), 16);
```

- **Convert.ToString**(данни) - от променлива към низ

```
string output = "Value: " + Convert.ToString(nums);
```

- **Convert.ToString**(данни, основа) – от променлива към число в бройна система със зададена основа. Числото се записва като низ

```
string output = "Binary Value: " + Convert.ToString(nums, 2);
```

# Какво научихме днес?

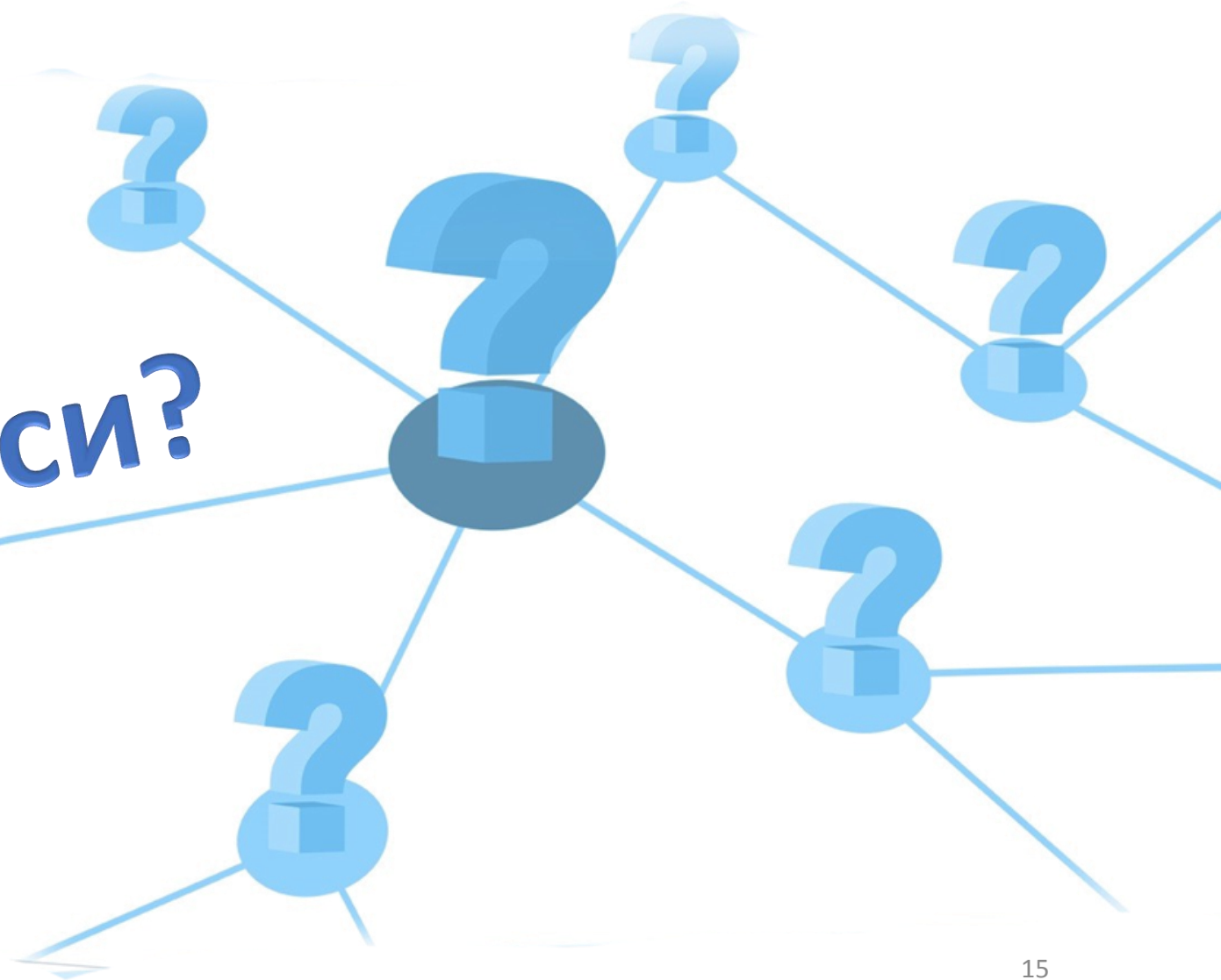
- Знаков тип: съдържа **Уникод** знак
- Низове: съдържат **текст**
  - Поредици от **Уникод** знаци
- Преобразуване на типове бива **скрито** и **явно**



Други типове данни



**Въпроси?**



# Договор за ползване

Този курс (слайдове, примери, задачи и др.) се разпространяват под свободен лиценз "[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International](https://creativecommons.org/licenses/by-nc-sa/4.0/)"



Базиран е на учебните материали на [НП „Обучение за ИТ Кариера“](#).

Може да съдържа части от следните източници:

- Книга "[Основи на програмирането със C#](#)" от Светлин Наков и колектив с лиценз [CC-BY-SA](#)