

Масиви

Работа с масиви: Постоянен размер.
Множества от елементи

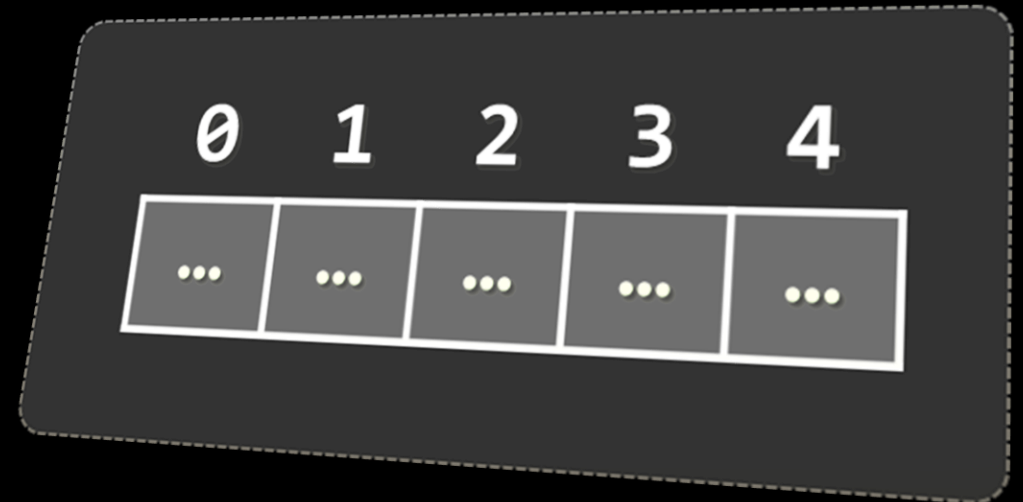
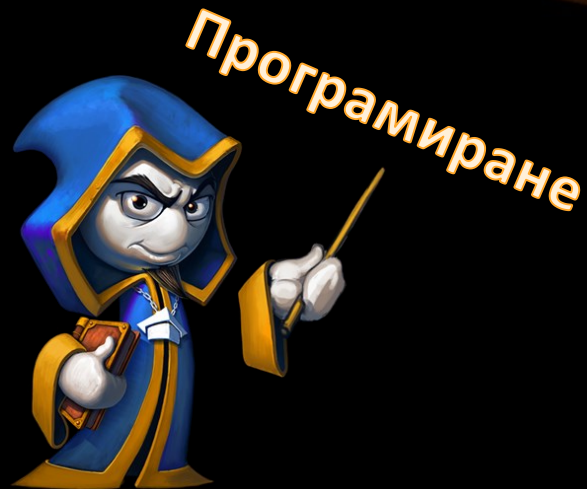


Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

<https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals>



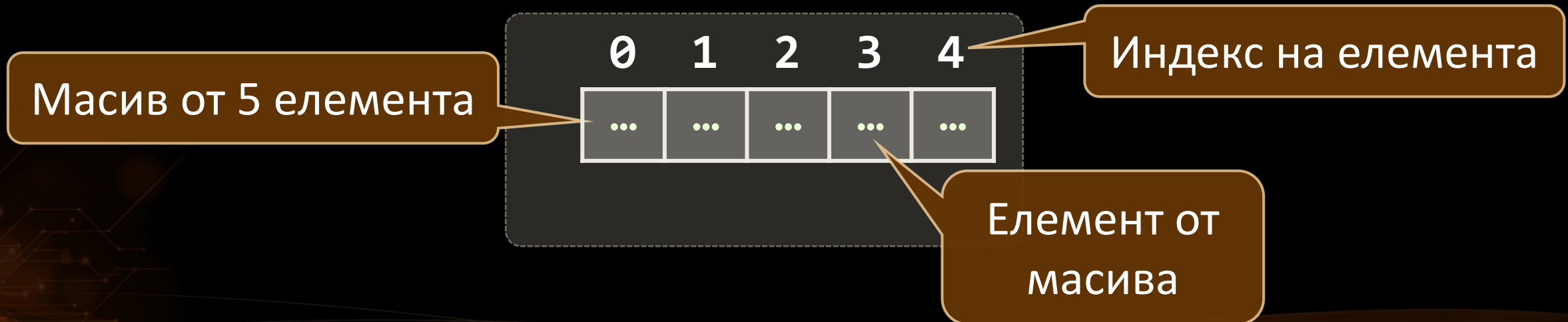
Съдържание

1. Що е масив?
2. Масиви от различни типове (примери)
3. Даване на стойност на елемент от масив
4. Достъп до елемент от масив
5. Стойностни типове данни
6. Референтни типове данни



Какво представляват масивите?

- В програмирането, масивът е множество от елементи
 - Елементите са номерирани от 0 до **Length-1**
 - Елементите са от **същия тип** (например integers – цели числа)
 - Масивите имат **постоянен размер(дължина) (Array.Length)** – не може да бъде променена



Работа с масиви

- Създаване на масив от 10 цели числа

Всички елементи
получават стойност == 0

```
int[] numbers = new int[10];
```

- Задаване на стойности на елементите на масива

В **Length** се пази дължината
(брой елементи) на масива

```
for (int i = 0; i < numbers.Length; i++)  
    numbers[i] = 1;
```

- Достъп до елементите на масива по индекс

Операторът **[]** дава
достъп до елементите
по **index**

```
numbers[5] = numbers[2] + numbers[7];  
numbers[10] = 1; // IndexOutOfRangeException
```


Дни от седмицата – Пример

- Дните от седмицата могат да бъдат запазени в **масив от низове**:

```
string[] days = {  
    "Monday",  
    "Tuesday",  
    "Wednesday",  
    "Thursday",  
    "Friday",  
    "Saturday",  
    "Sunday"  
};
```



Expression (име)	Value (Стойност)
days[0]	Monday
days[1]	Tuesday
days[2]	Wednesday
days[3]	Thursday
days[4]	Friday
days[5]	Saturday
days[6]	Sunday

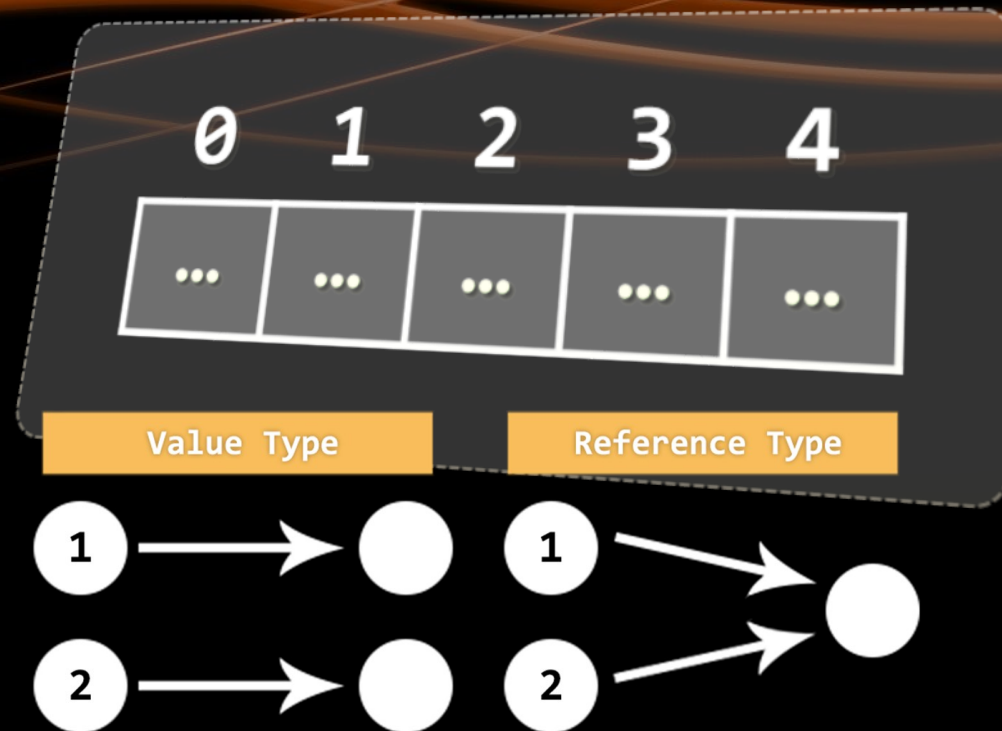
Задача: Ден от седмицата

- Въведете ден от седмицата като число [1...7] и изведете името на деня (in English) или "Invalid day!"

Locals			
Name	Value	Type	
days	{string[7]}	string[]	
[0]	"Monday"	string	
[1]	"Tuesday"	string	
[2]	"Wednesday"	string	
[3]	"Thursday"	string	
[4]	"Friday"	string	
[5]	"Saturday"	string	
[6]	"Sunday"	string	

Решение: Ден от седмицата

```
string[] days = { "Monday", "Tuesday", "Wednesday",  
"Thursday", "Friday", "Saturday", "Sunday" };  
int day = int.Parse(Console.ReadLine());  
  
if (day >= 1 && day <= 7)  
    Console.WriteLine(days[day - 1]);  
else  
    Console.WriteLine("Invalid day!");
```

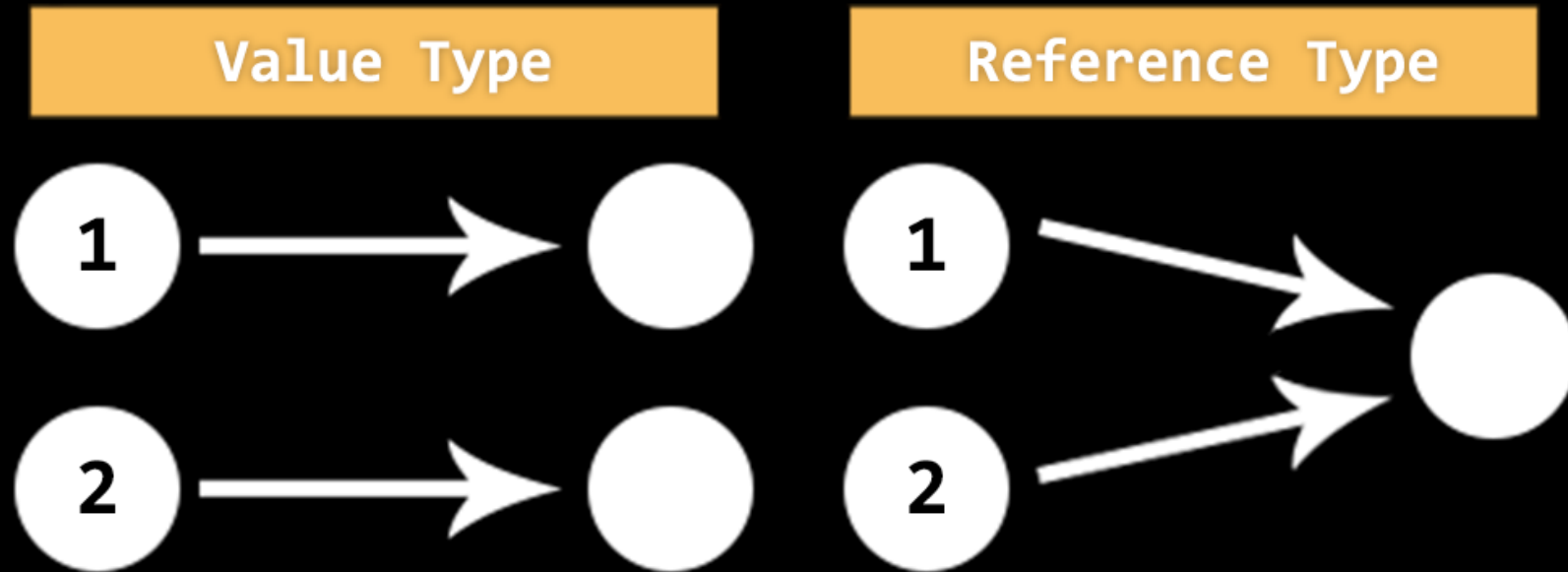


Достъп до елементите на масив

Референтни и стойностни типове

Стойностен и референтен тип

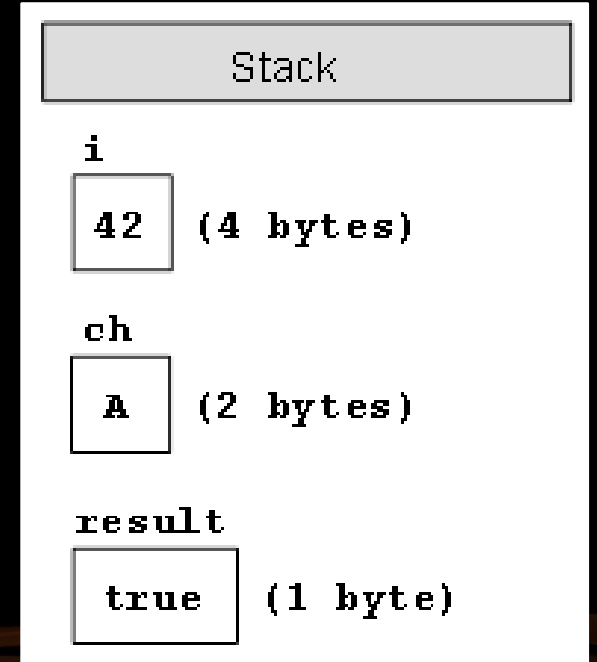
- Що е стойностен тип (Value Type)?
- Що е референтен тип (Reference Type)



Стойностни типове (Value Types)

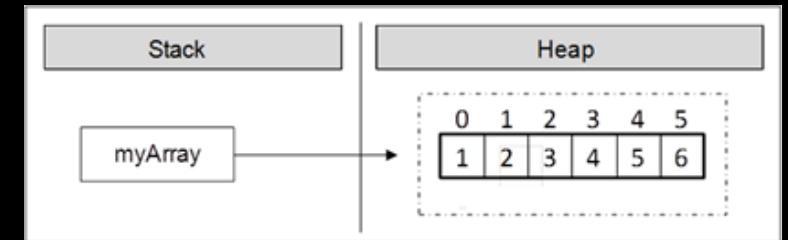
- **Стойностен тип** - променливите държат в себе си собствената стойност . В стека може да получим стойността на променливата като я извикаме по име
 - **int, float, double, bool, char, DateTime, BigInteger, ...**
 - msdn.microsoft.com/library/bfft1t3c.aspx
- Всяка променлива пази **копие** на **данните**

```
int i = 42;  
char ch = 'A';  
bool result = true;
```



Референтни типове(Reference Types)

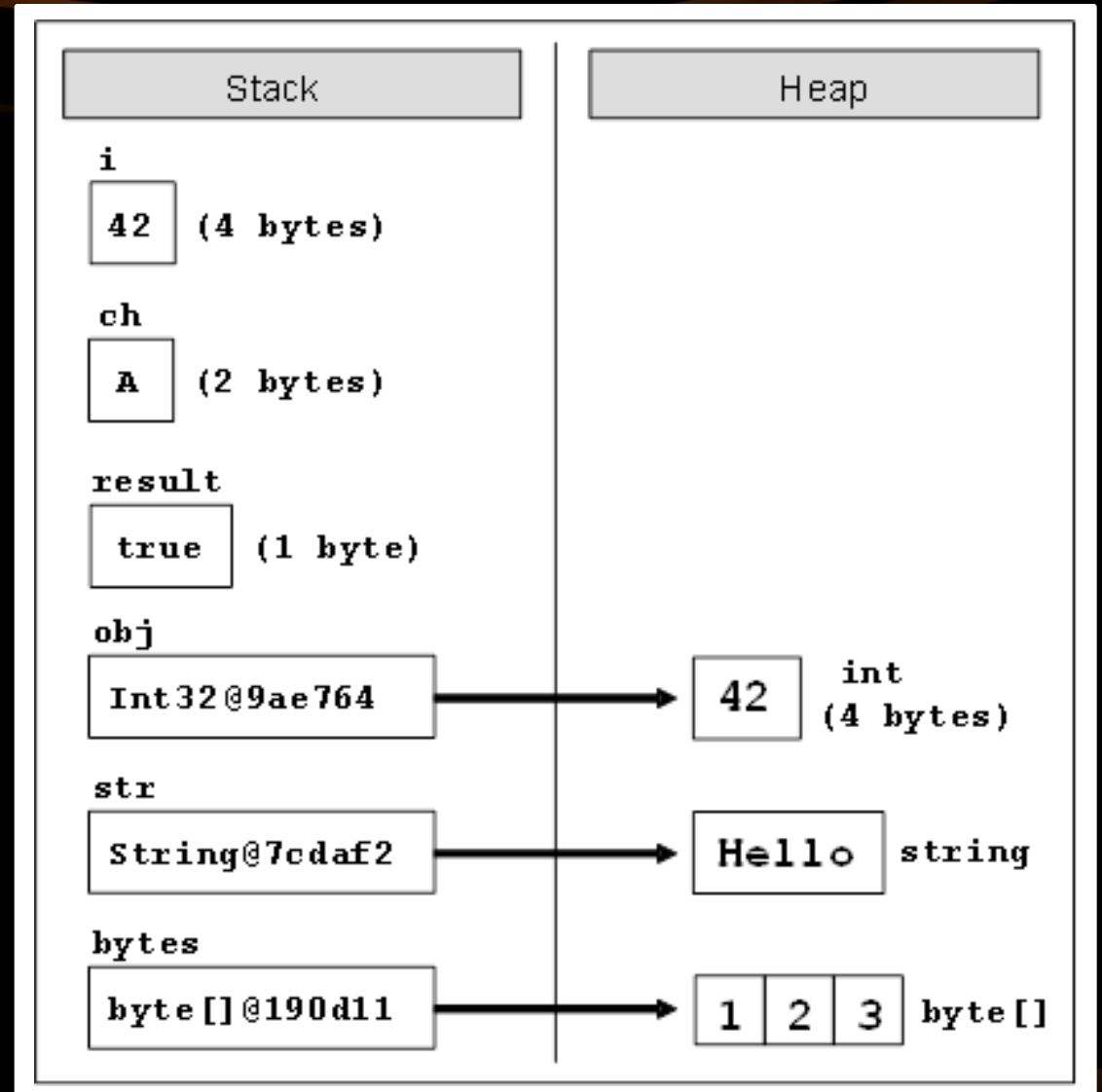
- Променливите от **референтен тип** съдържат (указател / адрес от паметта), на който се пазят стойностите на данните
 - **string, int[], char[], string[], Random**, инстанции на **classes, interfaces, delegates**
 - В този тип пазим не стойността, а **адреса** на стойността
- Две променливи от референтен тип могат да сочат (реферират) **един и същи обект**
 - Операциите за достъп/промяна чрез двата обекта въздействат върху едни и същи данни



```
var arr = new int[] { 1, 2, 3, 4, 5, 6 };
```

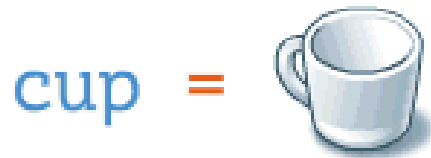
Разлики между референтен и стойностен тип

```
int i = 42;  
char ch = 'A';  
bool result = true;  
object obj = 42;  
string str =  
    "Hello";  
byte[] bytes =  
    { 1, 2, 3 };
```



Разлики между референтен и стойностен тип

pass by reference



fillCup()

pass by value



fillCup()

www.penjee.com

Пример: Стойностен тип (Value Types)

```
public static void Main()  
{  
    int num = 5;  
    Increment(num, 15);  
    Console.WriteLine(num);  
}
```

num == 5

```
private static void Increment(int num, int value)  
{  
    num += value;  
}
```

num == 20

Пример: Референтен тип (Reference Types)

```
public static void Main()  
{  
    int[] nums = { 5 };  
    Increment(nums, 15);  
    Console.WriteLine(nums[0]);  
}
```

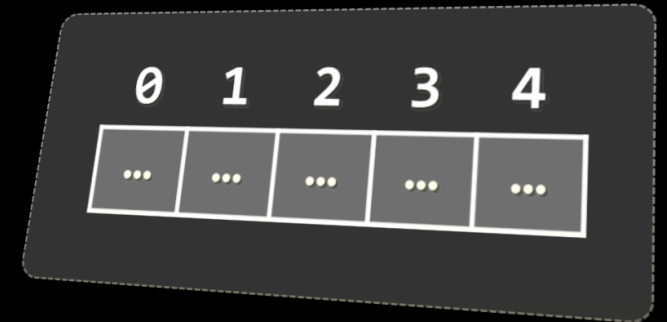
num == 20

```
private static void Increment(int[] nums, int value)  
{  
    nums[0] += value;  
}
```

num == 20

Какво научихме този час?

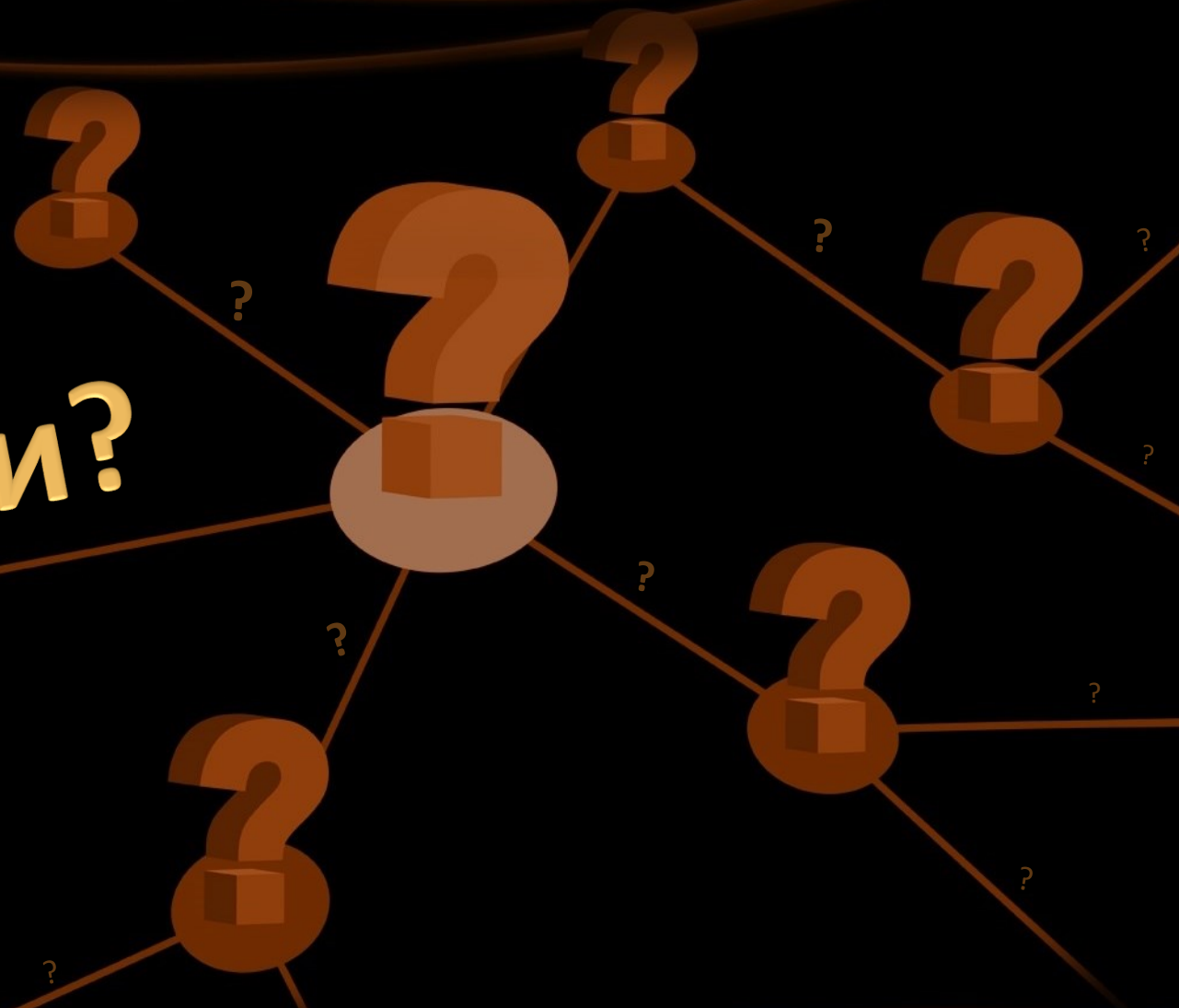
- Масивите са променливи от **един и същи тип**, с **едно и също име**, различаващи се по **индекс**
- Достъпът до елемент от масив става с посочване на **името на масива** и **индекса** му
- Индексите са от **0** до **Length-1**. Броят на елементите е **постоянен**
- Масивите са **референтен тип данни**, т.е. в тях се помни адреса, на който стоят данните, а не самите данни



Масиви



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

