

Масиви

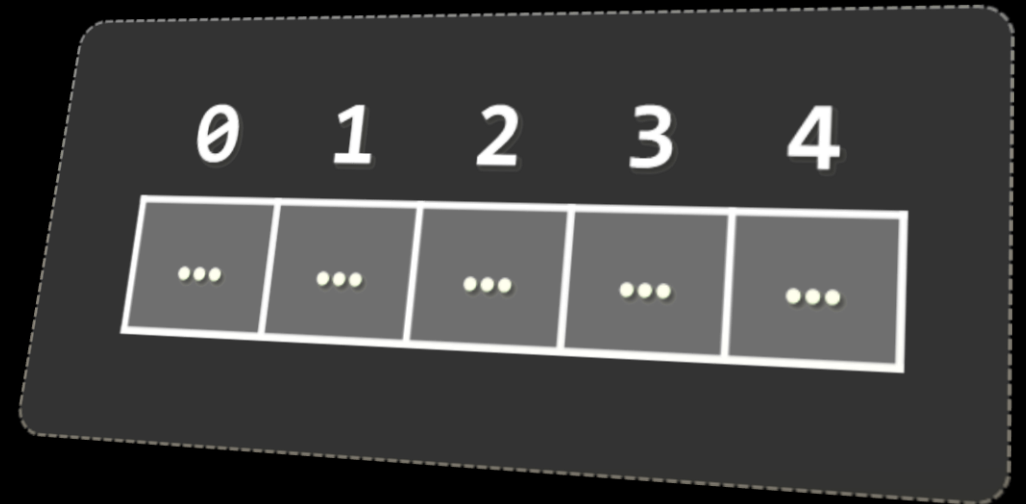
Достъп до елементите
Стойностни и референтни типове данни



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>



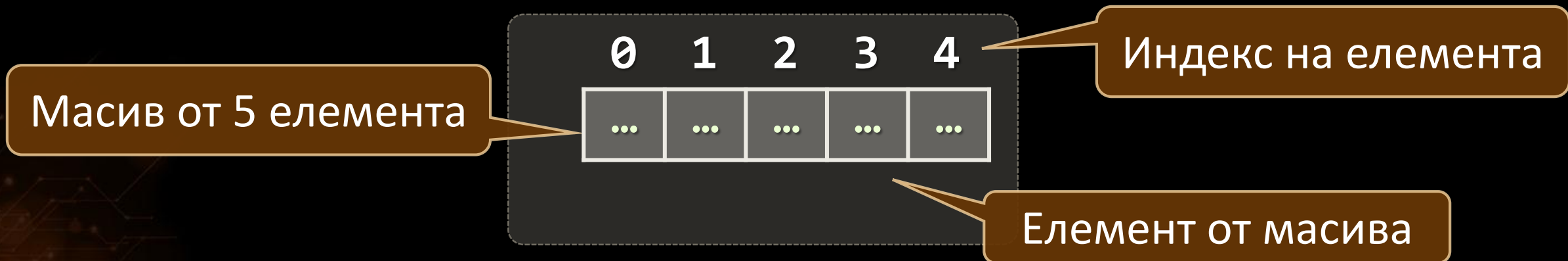
Съдържание

1. Що е масив?
2. Масиви от различен тип данни (примери)
3. Задаване 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 от **един и същ тип** (например `int` – цели числа)
 - Масивите имат **постоянен размер(дължина)** (**`Array.Length`**) – не може да се променя след създаване на масива
 - Елементите са **номерирани** от **0** до **`Length-1`**, това е техния **индекс**



Работа с масиви

0	1	2	3	4
...

- Създаване на масив от 10 цели числа:

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

Всички елементи
получават стойност 0

- Достъп до елементите е по техния индекс

```
numbers[1] = numbers[2] + numbers[3];  
numbers[10] = 1; // IndexOutOfRangeException
```

Операторът []
дава достъп
до елементите
по index

- Задаване на стойности на елементите на масива:

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

Length пази
дължината
(броя елементи)
на масива

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

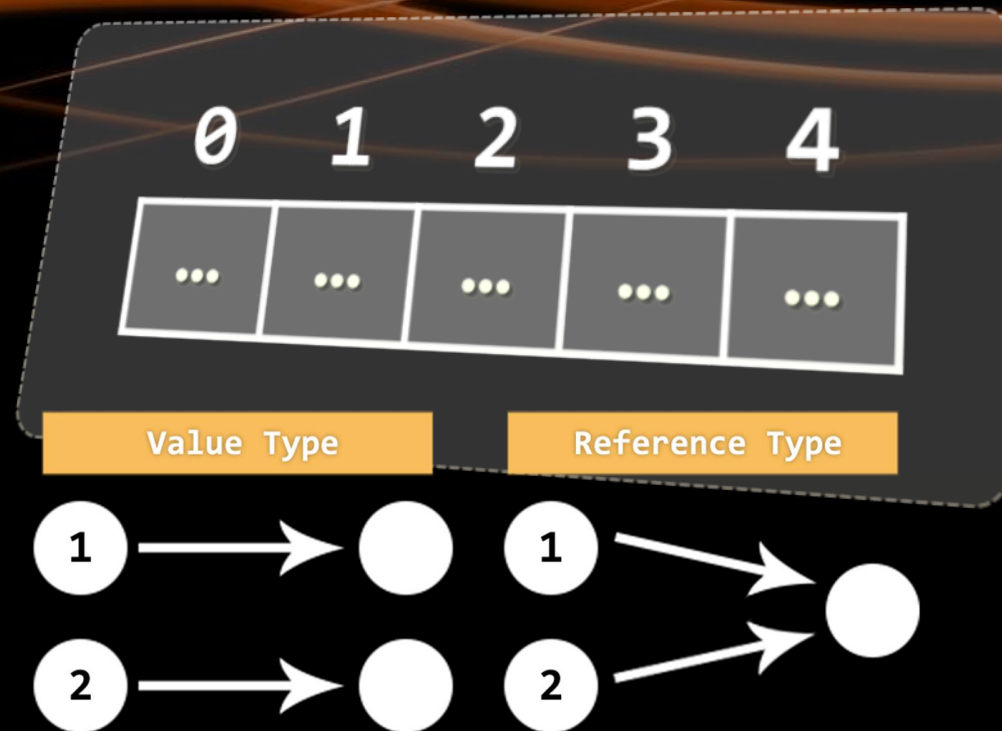
Задача: Ден от седмицата

- Въведете ден от седмицата като число [1...7] и изведете името на деня (на английски) 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!");
```

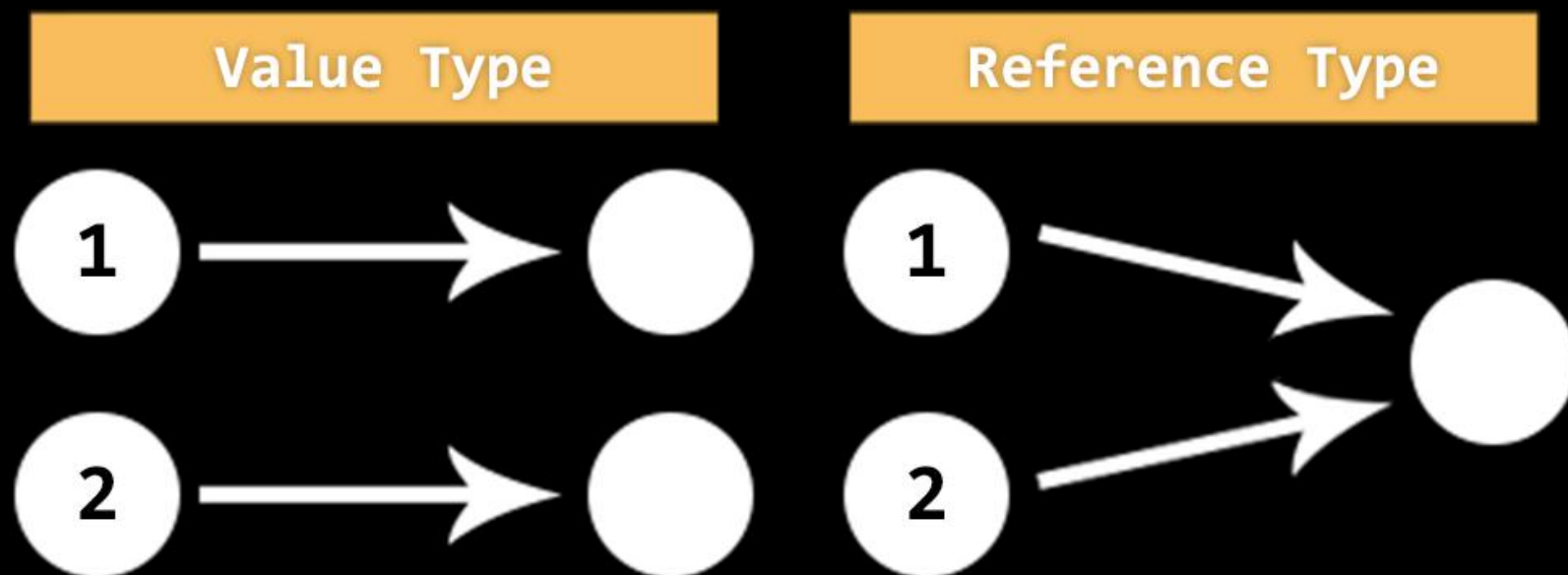


Достъп до елементите на масив

Референтни и стойностни типове

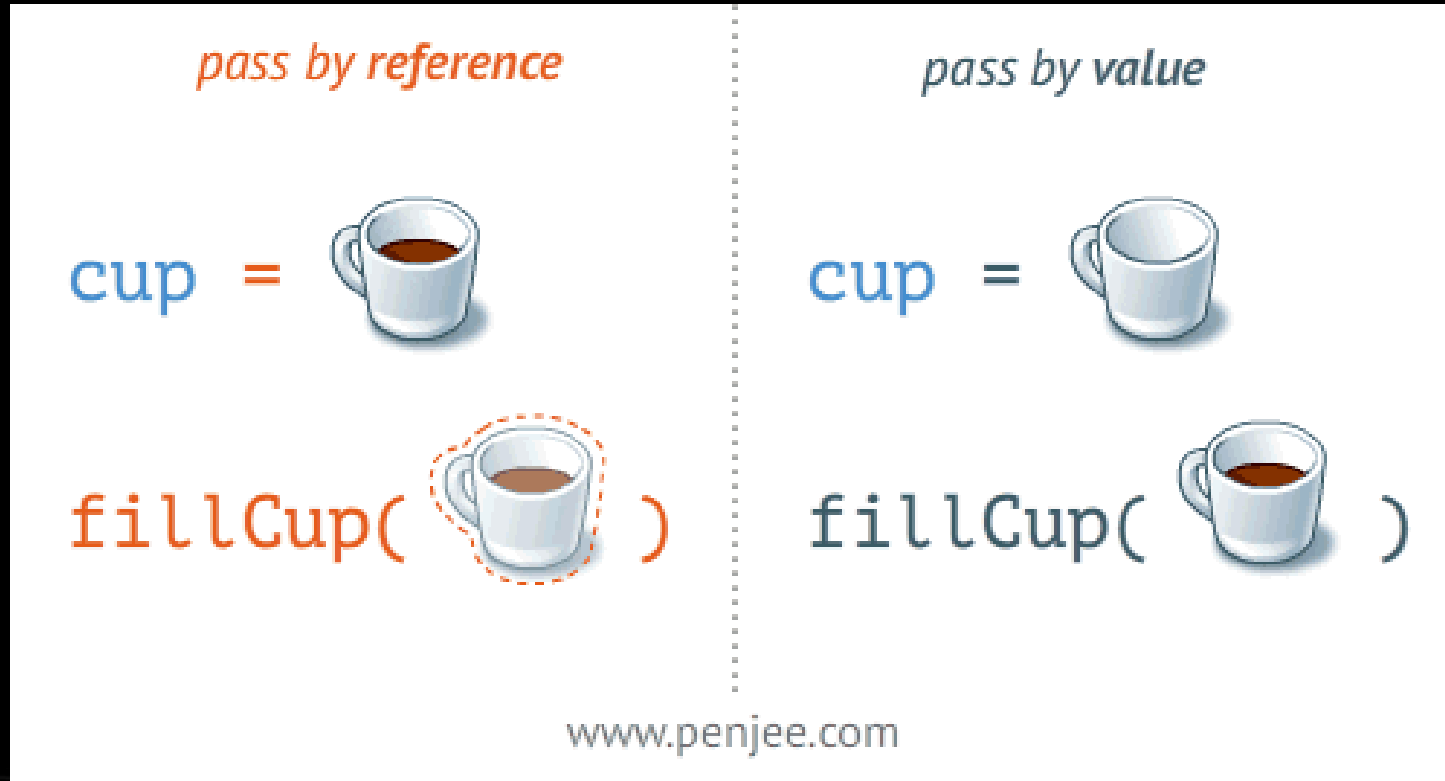
Стойностен и референтен тип

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



Разлики между референтен и стойностен тип

- Двата типа се държат много различно, когато са подадени като параметри на метод:



Пример: Стойностен тип (Value Types)

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

num == 5

5

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

num == 20

20

Пример: Референтен тип (Reference Types)

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

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

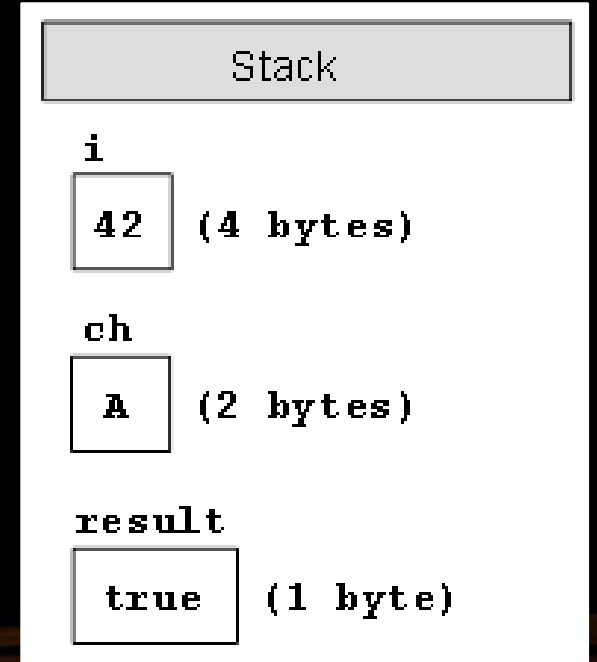
Diagram illustrating the state of memory for the reference type `int[]`:

- The `nums` array in `Main` and the `nums` parameter in `Increment` both point to the same memory location containing the value `20`.
- The callout `num == 20` next to `Console.WriteLine(nums[0]);` indicates that the value printed is `20`.
- The callout `num == 20` next to `nums[0] += value;` indicates that the value of the array element at index `0` is updated to `20` during the `Increment` operation.

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**

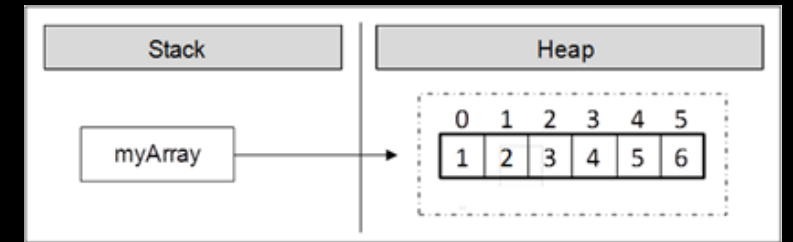
```
int i = 42;  
char ch = 'A';  
bool result = true;
```



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**

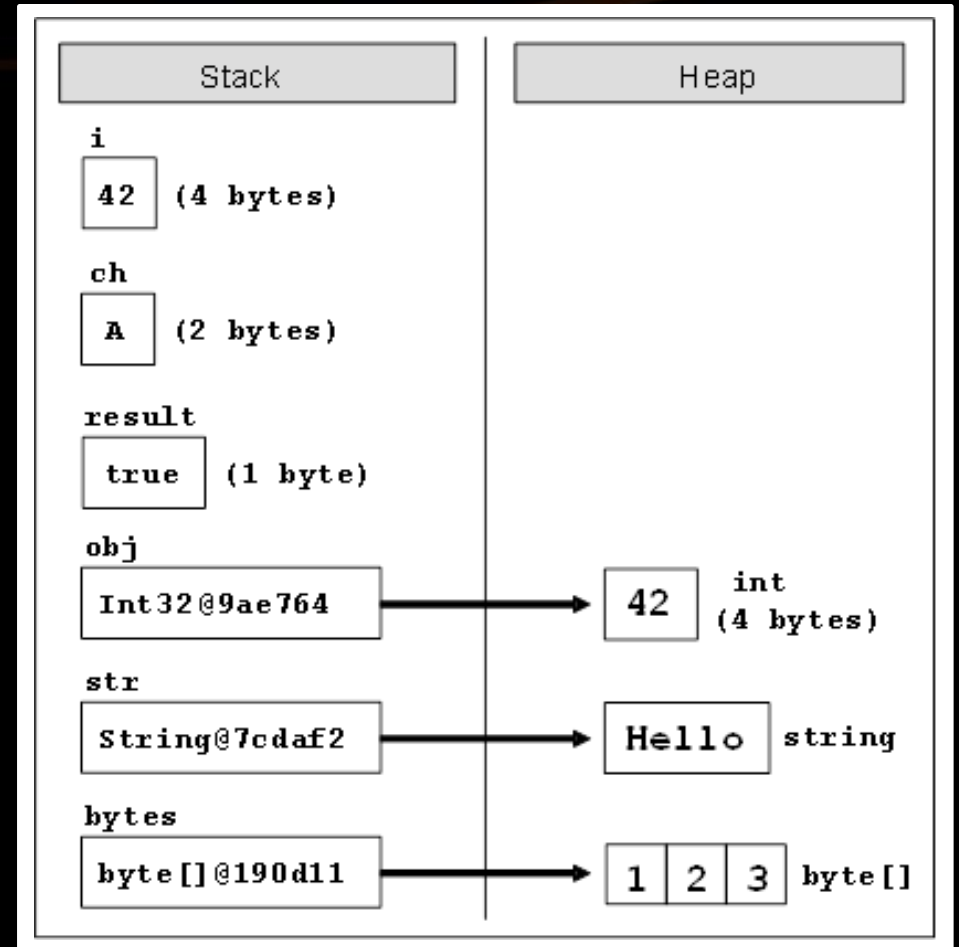
```
var arr = new int[] { 1, 2, 3, 4, 5, 6 };
```



- 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

Разлики между референтен и стойностен тип

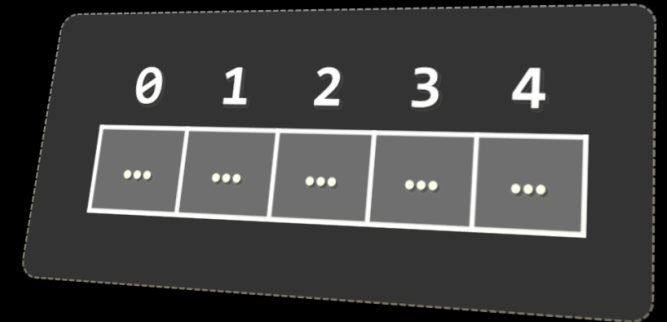
```
int i = 42;  
char ch = 'A';  
bool result = true;  
  
object obj = 42;  
string str = "Hello";  
byte[] bytes = { 1, 2, 3 };
```



Какво научихме този час?

- Массивите са множество променливи от един тип, с едно име, различаващи се по индекс
- Достъпът до елемент от масив става с посочване на името на масива и индекса на елемента
- Индексите са от 0 до Length-1. Броят на елементите е постоянен
- Массивите са референтен тип данни - променливите помнят адреса, на който стоят данните, а не самите данни

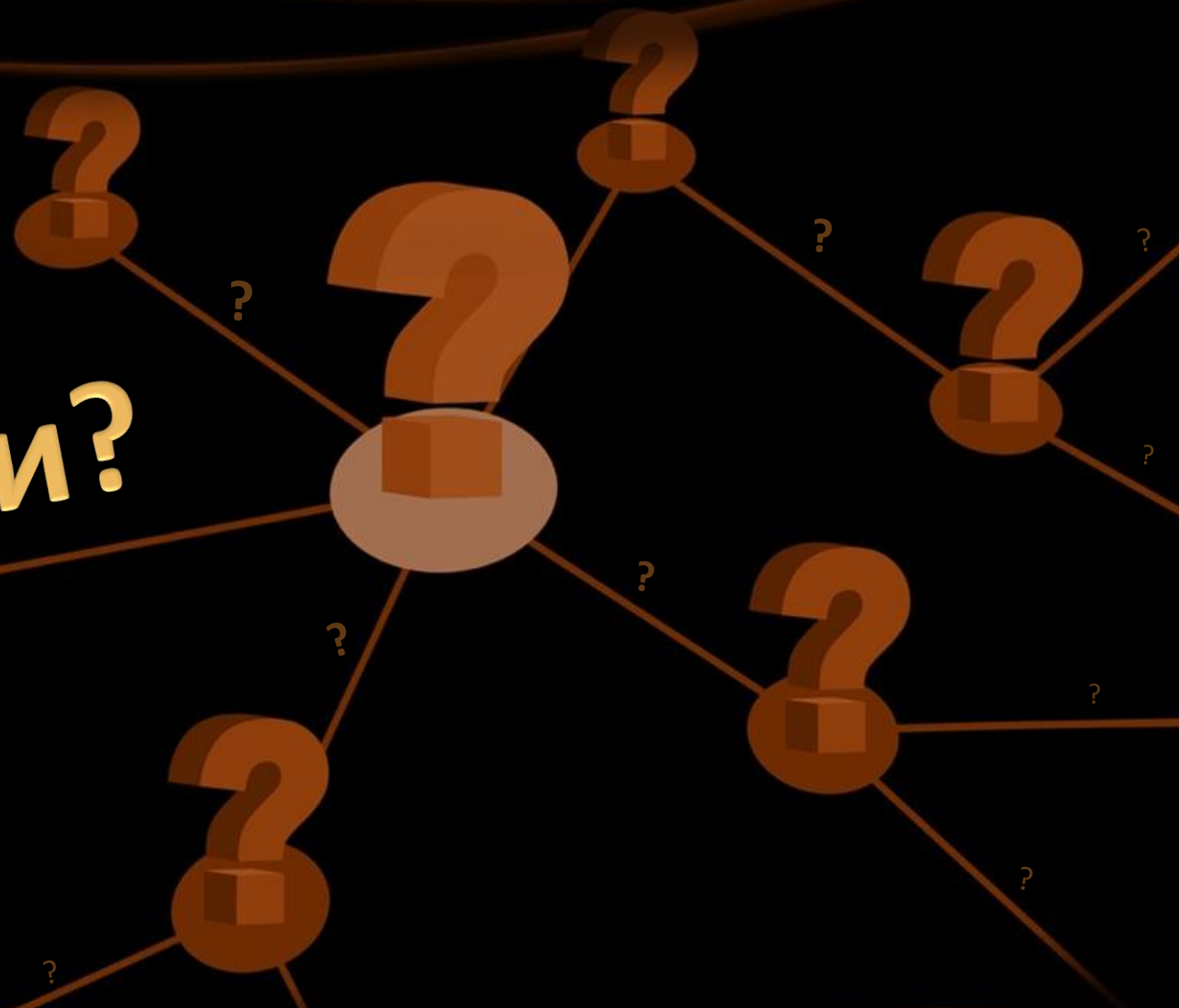
Summary



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

