

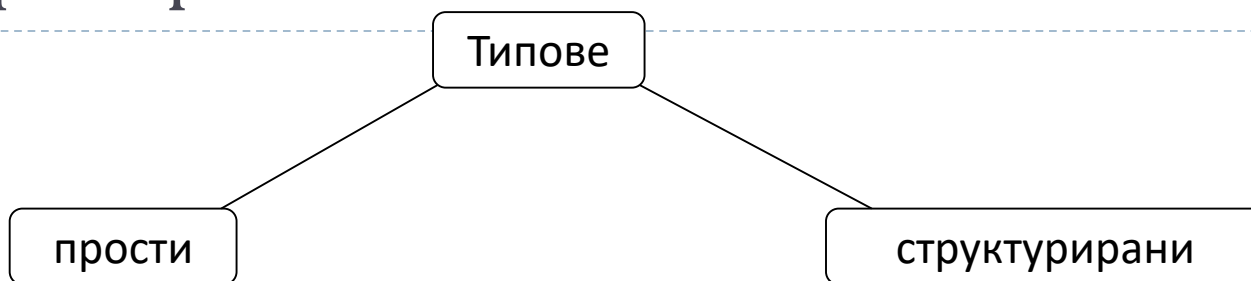
8. Едноразмерни масиви (Arrays)

Лекционен курс “Програмиране на Java”
проф. д-р Станимир Стоянов

Структура на лекцията

- ▶ Класификация на типовете
- ▶ Определение за масив и примери
- ▶ Декларация, създаване и достъп
- ▶ Работа с масиви
- ▶ Програмни параметри
- ▶ Примери
- ▶ Техники

Класификация на типовете в езиките за програмиране



прост - елементарен	съставен от други типове
<ul style="list-style-type: none">• Цели числа (byte, short, int, long)• Реални числа (float, double)• Boolean• Символи	<ul style="list-style-type: none">• Arrays (масиви) ¹⁾• Records (структури) ²⁾• Sets (множества) ³⁾• Pointer (указатели) ²⁾

- ¹⁾ в Java директно като тип
- ²⁾ в Java: чрез класове
- ³⁾ в Java:
 - собствена реализация
 - или: API-Class Set
 - или Java 5+: **enum** keyword

Arrays (масиви)

Последователност от стойности

- Еднакъв тип
- Непроменлива дължина

...

Защо ?

- Array от 'int'-стойности:

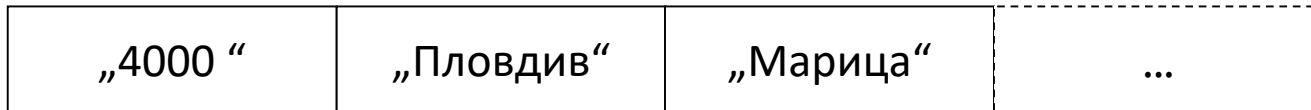
2	1000	-2	0
---	------	----	---

- Array от 'char'-стойности:

'a'	'b'	'e'	'r'
-----	-----	-----	-----

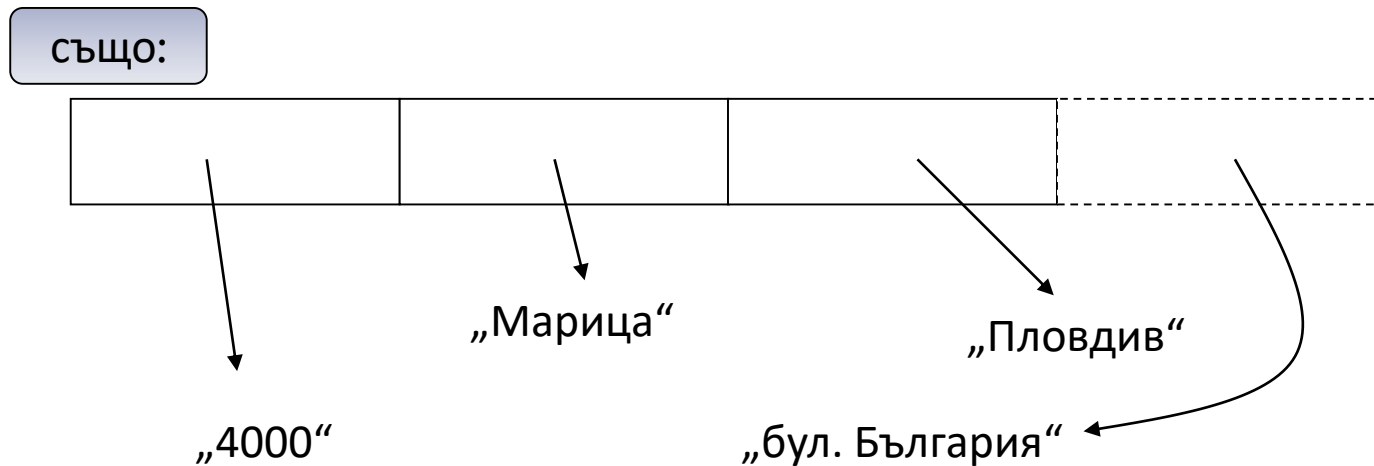
Array от ‚String‘–стойности:

Strings: непроменлива дължина?



Strings: обекти от един клас

→ представят се посредством адреси



Създаване на Arrays (1)

1. стъпка: декларация

```
double [] temperatures;  
String [] weekdays;
```

temperatures:

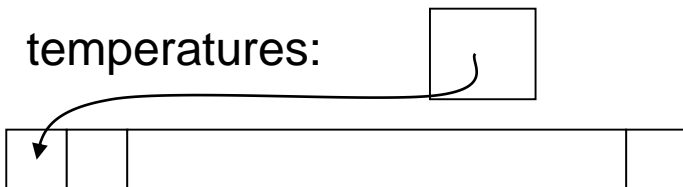


Единица памет!

2. стъпка: създаване

```
temperatures = new double [20];  
weekdays = new String [7];
```

temperatures:



От тук нататък
дължината е позната

Array-променливи са референции, т.е.
адреси → произволно големи Arrays

Създаване на Arrays (2)

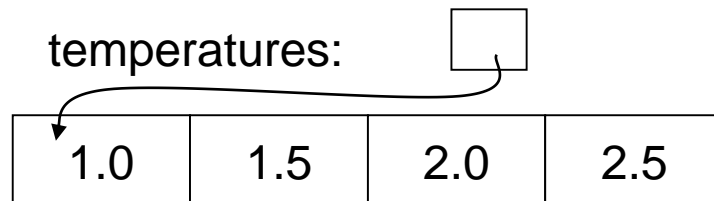
1. и 2. стъпки: заедно

(при декларации: брой на елементите е познат)

```
double [] temperatures  
    = new double [20];
```

Създаване посредством начална стойност

```
double [] temperatures  
    = {1.0, 1.5, 2.0, 2.5};
```



Arrays в Pascal

Декларациите и създаването не са разделени

```
VAR temperatures:  
    ARRAY [1900..2005] OF real;
```

семантични индексни области

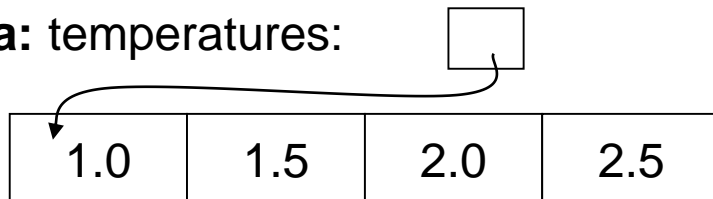
Arrays без референции

синтаксис четем

temperatures:

1.0	1.5	2.0	2.5
-----	-----	-----	-----

Java: temperatures:



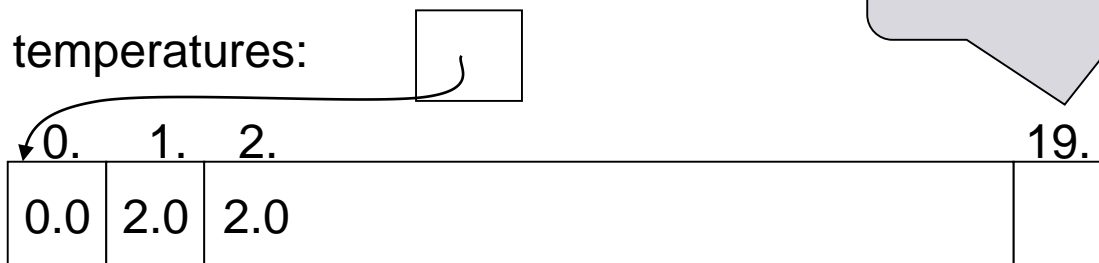
Достъп: отделни компоненти

```
double [] temperatures  
    = new double [20];
```

```
temperatures[0] = 0.0;  
temperatures[1] = 2.0;  
temperatures[2] =  
    temperatures[1];
```

Индексна област: 0 .. 19

temperatures:



Array: последователна структура с непроменлива дължина

Обработка: for-оператор

```
final int N = 101;  
int [] arr = new int[N];  
  
for (int i=0; i < N; i++)  
    // i: 0 .. 100  
    arr[i] = -i;
```

run-time грешка
Каква?

Може ли $i \leq N$?

Arrays: обработка

Java:

```
for (int i = 0; i < N; i++)  
    arr[i] = -i;
```

Pascal:

```
for i := 0 to N - 1 do  
    arr[i] := -i;
```

Методи: за Arrays с произволна дължина

Метод: сума на елементите за произволно дълги Arrays

```
int sum (int[] vector) {  
    int total = 0;  
  
    for (int i = 0; i < vector.length; i++)  
        total += vector[i];  
    return total;  
}
```

Приложение:

```
int [] v1 = new int[8];  
int [] v2 = new int[100];  
    . . . // v1, v2: попълване  
s1 = sum(v1);  
s2 = sum(v2);
```

Техниката е възможна,
понеже Arrays в Java
са референции

Програмни параметри

```
class Temperature {  
    // Преобразуване на температура  
    // от Фаренхайт в Целзий  
  
    public static void main (String[] args) {  
        double tempFahr; // Фаренхайт  
        double tempCels; // Целзий  
        System.out.print("Temperature (deg F): ");  
        tempFahr = Keyboard.readDouble();  
        tempCels = (5.0 * (tempFahr - 32.0)) / 9.0;  
        System.out.print(tempFahr);  
        System.out.print(" deg F is ");  
        System.out.print(tempCels);  
        System.out.println(" deg C");  
    }  
}
```

Формални параметри:
- използване ?

Актуален параметър?

Извикване от main ?

```
% javac Temperature.java  
% java Temperature
```

```
Temperature (deg F): 10  
10 deg F is -12.222222222222221 deg C
```

Програмни параметри

```
class select {  
    public static void main (String[] args) {  
        ...  
    }  
}
```

Формален параметър от тип 'Array' от 'String'

Актуален параметър ?

```
% java select -p1-5 file.ps
```

args:



Параметър от командния ред

-p1-5

file.ps

= актуален параметър

Програма: управлявана от параметри при нейното извикване

```
% java temp -C -F,K  
% java temp -K -C,K
```

Java-програми

Достъпна ли е args в програмата?

args:

-C -F,K

Променливата args може да бъде анализирана от програмата

Пример: програмни параметри

```
public class Echo {  
    public static void main(String args[]) {  
        for (int i = 0; i < args.length; i++)  
            System.out.print(args[i] + "\n");  
    }  
}
```

```
% java Echo -p1-5,6 f1.ps f2.ps  
-p1-5,6  
f1.ps  
f2.ps
```

args:

-p1-5,6	f1.ps	f2.ps
---------	-------	-------

3 програмни параметри
→ 1 актуален параметър от тип Array с дължина 3

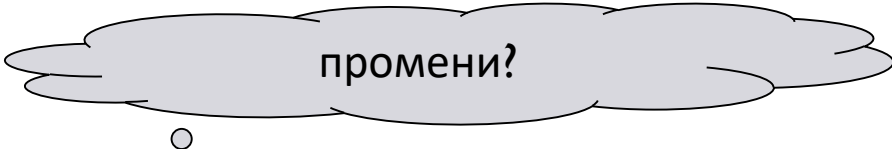
Ехо

досега:

```
% java Echo -p1-5,6 f1.ps f2.ps  
-p1-5,6  
f1.ps  
f2.ps
```

НОВО:

```
% java Rev -p1-5,6 f1.ps f2.ps  
f2.ps f1.ps -p1-5,6
```



промени?

```
public static void main(String args[]) {  
    for (int i=args.length-1; i >= 0; i--)  
        System.out.print(args[i] + " ");  
}
```

Boolean-Arrays

Пример: Намерете простите числа до определена граница.

- **Техника:** мрежа на Ератостен

- **Основна идея:**

Задраскваме всички кратни на вече разпознатите прости числа.

```
boolean[] net;
```

```
// Позицията в масива съответства на числото  
// net[i] = true <-> i е просто число
```

0	1	2	3	4	5	6	(Index)
false	false	true	true	true	true	...	

Начално присвояване

Prime number програма

- Изход: крайният изход на получените прости числа
- net[0], net[1]?

```
boolean[] net;  
int i, j, n;
```

декларация

```
System.out.print("Prim number border: ");
```

```
n = Keyboard.readInt();
```

```
// now: memory requirement
```

```
net = new boolean[n];
```

създаване

```
// all are potential prim numbers:
```

```
for (i = 2; i < n; i++)
```

```
    net[i] = true;
```

```
for (i = 2; i < n; i++)
```

```
    if (net[i]) { // if i prim number,
```

```
        // delete all repeatable of i
```

```
        for (j = i+i; j < n; j += i)
```

```
            net[j] = false;
```

```
}
```

Константни Arrays, паралелни Arrays

Пример:

Управление на дати и месеци:

- Имена
- Брой дни

```
final String[] MONTH_NAME = { "",  
    "януари", "февруари", "март",  
    "април", "май", "юни",  
    "юли", "август", "септември",  
    "октомври", "ноември", "декември" };  
  
final int[] DAYS_OF_MONTH = { 0,  
    31, 28, 31, 30,  
    31, 30, 31, 30,  
    30, 31, 30, 31};
```

Array-индекс: доставя кореспондиращата двойка данни

```
final String[] MONTH_NAME = { "",  
    "януари", "февруари", "март",  
    "април", "май", "юни",  
    "юли", "август", "септември",  
    "октомври", "ноември", "декември"};
```

```
final int[] DAYS_OF_MONTH = { 0,  
    31, 28, 31, 30,  
    31, 30, 31, 30,  
    30, 31, 30, 31};
```

```
final int MAY = 5;
```

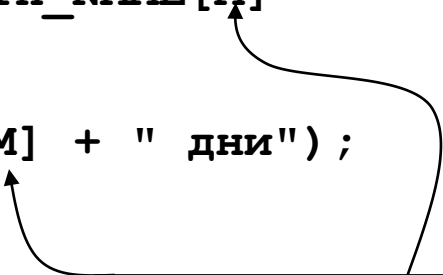
Защо ?

	1	2	3	4	5	6	7
MONTH_NAME:	"януари"	"февруари"	"март"	"април"	"май"	"юни"	
DAYS_OF_MONTH:	31	28	31	30	31	30	

Array-индекс: доставя кореспондиращата двойка данни

```
static int M = MAY;

public static void main (String[] args) {
    System.out.println(
        "Месец " + MONTH_NAME[M]
        + " има " +
        DAYS_OF_MONTH[M] + " дни");
}
```



Същият индекс

Изход:

Месец май има 31 дни

Благодаря за вниманието!

Край лекция 8. “Едноразмерни масиви (Arrays)”