

انجمن جاوا کاپ تقدیم می کند

دوره برنامه نویسی جاوا

رشته، آرایه و چند داستان دیگر

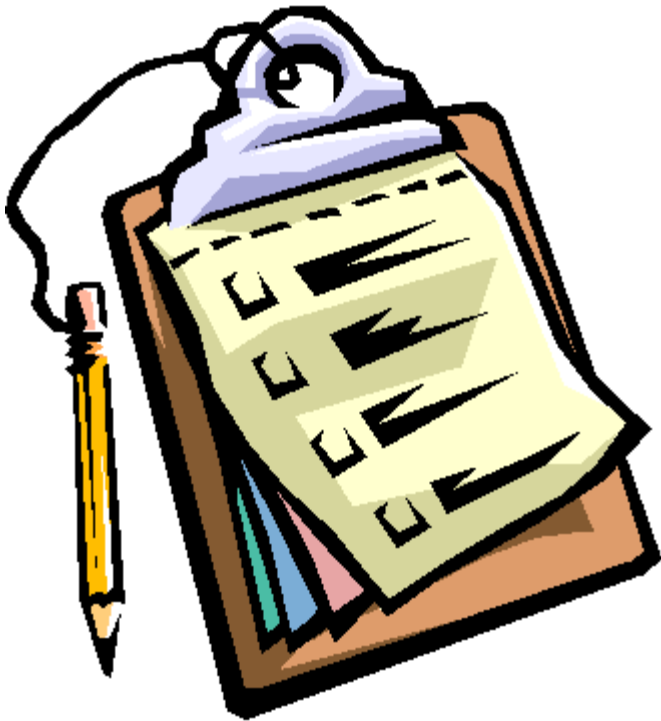
STRING, ARRAY, AND OTHER STORIES

صادق علی اکبری

- کلیه حقوق این اثر متعلق به انجمن جاواکاپ است
- باز نشر یا تدریس آن چه توسط جاواکاپ و به صورت عمومی منتشر شده است، با ذکر مرجع (جاواکاپ) بلامانع است
- اگر این اثر توسط جاواکاپ به صورت عمومی منتشر نشده است و به صورت اختصاصی در اختیار شما یا شرکت شما قرار گرفته، باز نشر آن مجاز نیست
- تغییر محتوای این اثر بدون اطلاع و تأیید انجمن جاواکاپ مجاز نیست



سرفصل مطالب



- ورودی کاربر

- Scanner

- کنترل نوع داده در جاوا

- سایر ساختارهای کنترل جریان

- switch

- break & continue

- رشته

- آرایه





برنامه‌های کنسول

Console Applications

برنامه‌های کنسول

- برنامه‌هایی که مبتنی بر ورودی و خروجی متنی هستند

- در یک کنسول اجرا می‌شوند

- مثلاً cmd در ویندوز و shell در لینوکس

- ورودی و خروجی استاندارد

- Standard input

- ورودی با کمک صفحه کلید دریافت می‌شود

- Standard Output

- خروجی به صورت متنی در کنسول چاپ می‌شود

- ما فعلاً برنامه‌ها را به صورت کنسولی می‌نویسیم

- انواع دیگر برنامه‌ها بر مبنای همین اصول ساخته می‌شوند

- Web applications, Desktop Applications

```
Welcome to Product Database Application
1. Add Product
2. Edit Product
3. View Product
4. Delete Product
5. Exit

Enter your Selection:
```



دریافت مقدار از ورودی

- می‌دانیم که با کمک `println` خروجی چاپ می‌شود

`System.out.println` •

- چگونه از ورودی استاندارد استفاده کنیم؟

- راه ساده: استفاده از `Scanner` و `System.in`

- مثال:

```
Scanner scanner = new Scanner(System.in);
```

```
int n = scanner.nextInt();
```

```
double d = scanner.nextDouble();
```

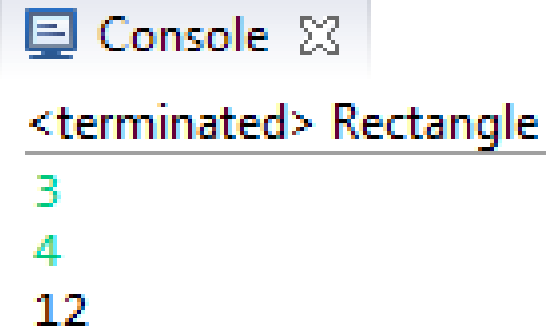
- برای استفاده از `Scanner`، در اوایل برنامه عبارت زیر را وارد کنید:

```
import java.util.Scanner;
```



```
import java.util.Scanner;

public class Rectangle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int width = scanner.nextInt();
        int length = scanner.nextInt();
        long area = width*length;
        System.out.println(area);
        scanner.close();
    }
}
```



```
<terminated> Rectangle
3
4
12
```



کنترل نوع

Type Checking

کنترل نوع

- جاوا نوع داده‌ها را به دقت کنترل می‌کند
- از برخی انتساب‌ها جلوگیری می‌کند

```
int intVal = 2;
```

```
long longVal = 12;
```

```
intVal = longVal; ❌ Syntax Error
```

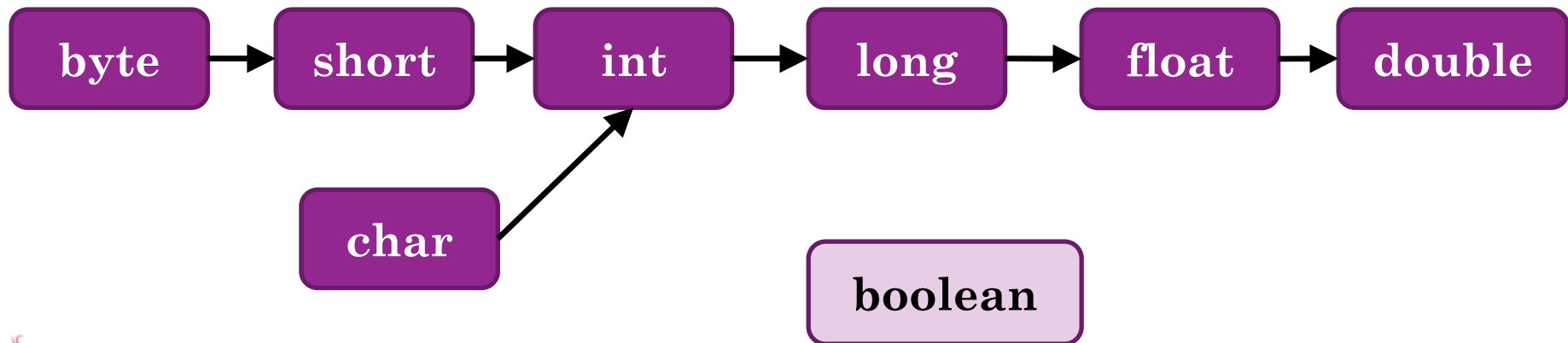
```
longVal = intVal; ✅ OK
```

```
intVal = (int)longVal; ✅ OK (Type Casting)
```



تبدیل مستقیم انواع داده

- تبدیل نوع در جهت فلش‌ها ممکن است
- همه تبدیل‌های دیگر نیازمند تبدیل صریح (Cast) هستند
- نوع داده boolean قابل تبدیل نیست
- نوع داده char یک نوع ویژه عددی است



جدول تبدیل نوع داده‌ها

Convert From:	Convert To:							
	boolean	byte	short	char	int	long	float	double
boolean	-	N	N	N	N	N	N	N
byte	N	-	Y	C	Y	Y	Y	Y
short	N	C	-	C	Y	Y	Y	Y
char	N	C	C	-	Y	Y	Y	Y
int	N	C	C	C	-	Y	Y*	Y
long	N	C	C	C	C	-	Y*	Y*
float	N	C	C	C	C	C	-	Y
double	N	C	C	C	C	C	C	-

● N : تبدیل ممکن نیست

● C : نیاز به تبدیل صریح (Cast)

● Y : تبدیل به صورت خودکار و

● Y* : تبدیل خودکار است ولی دقیق

ضمنی توسط جاوا انجام می‌شود

نیست



```
int i;    float f;
```

```
i = 123456789;    یک عدد صحیح بزرگ
```

```
f = i;    تخمینی از مقدار i در f ذخیره می‌شود
```

```
System.out.println(f);    خروجی: 1.23456792E8 (همان عدد صحیح بزرگ نیست)
```

```
i = (int) f;
```

```
System.out.println(i);    خروجی: 123456792
```

- انواع داده اعشاری (float و double)، عملاً برآوردی تقریبی از اعداد را نگه می‌دارند



چند نکته درباره اعداد اعشاری

- مقدار مبهم (Double.NaN)

- `double nan = 0.0/0.0;`

- مقدار بینهایت (Double.NEGATIVE_INFINITY)

- `double inf = Double.MAX_VALUE*2;`

- بینهایت منفی (Double.POSITIVE_INFINITY)

- `double inf = Double.MAX_VALUE*(-2);`

- نمایش مناسب اعداد اعشاری (Formatting)

- `System.out.format("min double = %5.2f", fvariable);`



مقایسه اعداد اعشاری

- مقایسه مستقیم اعداد اعشاری
- استفاده از عملگر `==` یا `!=` برای مقایسه اعداد اعشاری مناسب نیست (اشتباهی رایج است)
- مثال: یک حلقه بی نهایت (چرا این حلقه تمام نمی‌شود؟)

```
for (float f = 10f; f != 0; f -= 0.1) {  
    System.out.println(f);  
}
```

- روش بهتر:

```
for (float f = 10f; f > 0; f -= 0.1) {  
    System.out.println(f);  
}
```



انتساب‌های عددی

- سرریز انتساب اعداد صحیح:

- با انتساب یک عدد صحیح بزرگ در یک عدد صحیح از نوع ضعیف‌تر، خطای زمان اجرا رخ نمی‌دهد
- بلکه خطای منطقی رخ می‌دهد (فقط بیت‌های کم‌اهمیت‌تر استفاده میشوند)، مثال:

```
long l = 123456789_123456789L;
```

```
int i = (int) l;
```

```
System.out.println(i); خروجی: -1395630315
```

- سرریز انتساب اعداد اعشاری در اعداد صحیح: حداکثر عدد ممکن ذخیره می‌شود، مثال:

```
double d = 123456789_123456789D;
```

```
int i = (int) d;
```

```
System.out.println(i);
```

```
System.out.println(Integer.MAX_VALUE);
```

خروجی:

2147483647

2147483647





سایر ساختارهای کنترل جریان برنامه

Flow Control Structures

ساختار SWITCH

```
switch(x){  
case a: folan();break;  
case b: bahman();break;  
...  
}
```

- جایگزینی برای ساختار if-else با ساختمانی بهتر

- کاربرد: وقتی که شرطها، بررسی تساوی یک مقدار با گزینه‌های مختلف است

- اگر x برابر با a بود فلان کار را بکن، اگر x مساوی b بود بهمان کار را بکن و ...

- انواع داده‌ای که در switch قابل استفاده هستند:

- اعداد صحیح (int) و انواع داده قابل تبدیل به int

- int, byte, short, char

- در مثال فوق x نمی‌تواند از این انواع باشد: long, double, float, boolean

- رشته (String) و enum

- رشته: از نسخه ۱.۷ به بعد

درباره String و enum
بعداً صحبت خواهیم کرد



مثالی برای SWITCH

```
switch (i) {  
case 1:  
    System.out.println("YES");  
    break;  
case 2:  
    System.out.println("NO");  
    break;  
default:  
    System.out.println("Perhaps");  
}
```

معادل همین switch-case با کمک if-else :

```
if(i==1)  
    System.out.println("Yes");  
else if(i==2)  
    System.out.println("No");  
else  
    System.out.println("Perhaps");
```



```
Scanner scanner = new Scanner(System.in);
boolean again = true;
while (again) {
    System.out.println("1: Play");
    System.out.println("2: Setting:");
    System.out.println("3: Exit");
    System.out.print("Enter Your Choice:");
    int i = scanner.nextInt();
    switch (i) {
        case 1:
            play();
            break;
        case 2:
            setting();
            break;
        case 3:
            again = false;
            break;
        default:
            System.out.println("Enter a valid number");
    }
}
```

مثالی دیگر برای SWITCH

دستور BREAK

- این دستور اجرای یک حلقه را متوقف می کند
- برنامه از حلقه خارج می شود

```
while (true) {  
    int nextInt = scanner.nextInt() ;  
    if (nextInt == 0)  
        break ;  
    . . .  
}
```



دستور CONTINUE

- این دستور یک گام از اجرای حلقه را لغو می‌کند
- گام بعدی را از ابتدای حلقه ادامه می‌دهد.

```
for (int i=0; i<10; i++) {  
    if (i==4) continue;  
    System.out.println(i);  
}
```

- نکته: تفاوت دستور continue در دو حلقه for و while



حلقه‌های تودرتو

```
Scanner scanner = new Scanner (System.in);  
int nextInt;  
do{  
    nextInt = scanner.nextInt() ;  
    for(int i=0;i<nextInt;i++){  
        System.out.println(i);  
    }  
}while (nextInt>0) ;
```

- چگونه دستور break یا continue را برای یک حلقه خارجی تر اعمال کنیم؟



● استفاده از برچسب برای حلقه‌ها

```
outer: for (int i = 0; i < 10; i++)  
    inner: for (int j = 0; j < 10; j++) {  
        if (j == 2)  
            break outer;  
        else {  
            System.out.println(i);  
            System.out.println(j);  
            continue inner;  
        }  
    }  
}
```



نکته روز

Tip of the Day

نکته روز: دندانه گذاری (INDENTATION)

```
int nextInt;  
  
do{  
  
    nextInt = scanner.nextInt() ;  
  
    for(int i=0;i<nextInt;i++) {  
  
        System.out.println(i) ;  
  
    }  
  
}while (nextInt>0) ;
```



نکته روز: دندانه‌گذاری (INDENTATION)

```
int nextInt;  
  
do{  
  
    nextInt = scanner.nextInt() ;  
  
    for(int i=0;i<nextInt;i++) {  
  
        System.out.println(i) ;  
  
    }  
  
}while (nextInt>0) ;
```



توضیحات (COMMENTS)

- خطوطی که توسط کامپایلر نادیده گرفته می‌شود
- مفید برای اضافه کردن توضیحات
- و یا حذف موقت بخشی از برنامه (بهتر است حجم این نوع کامنت کم باشد)

● کامنت یک خطی: `//This is a temporary variable`

● کامنت چندخطی:

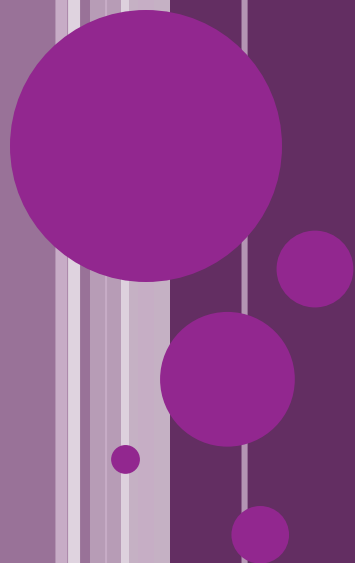
```
/*nextInt = scanner.nextInt();  
for(int i=0;i<nextInt;i++){  
    System.out.println(i);  
} */
```



مثال از کامنت

```
public boolean isPrime(int number) {  
    if(number <1)  
        return false;  
    /*if(isEven(number))  
        return false;  
    */  
    for(int i=2;i<number/2;i++) //searching for a divisible  
        if(number%i==0)  
            ...  
}
```





String

رشته

- رشته: مجموعه‌ای از کاراکترها (مثلاً نام یک فرد)
- کاراکتر (مقادیر ثابت رشته‌ای با ' شروع و تمام می‌شوند)
 - `char ch = ' a ';`
 - `char ch = ' 1 ';`
 - `char ch = ' # ';`
- رشته (مقادیر ثابت رشته‌ای با " شروع و تمام می‌شوند)
 - `String st = "Ali";`
 - `String st = "Salam123OK";`
 - `String st = "1";`
 - `String st = "";`



```
Scanner scanner = new Scanner(System.in);
String input;
input = scanner.next();
switch (input) {
case "Salam":
    System.out.println("Hi!"); break;
case "Khdahafez":
    System.out.println("Bye!"); break;
default:
    System.out.println("Ha?!"); break;
}
System.out.println(input);
```



- رشته یک نوع داده اولیه (Primitive Data Type) نیست
- برای کار با رشته‌ها، متدهایی تعریف شده است:

```
String input = "Ahay Salam Ahay Salam!!!";
```

```
char ch = input.charAt(0); A
```

```
int i = input.indexOf("Salam"); 5
```

```
int j = input.lastIndexOf("Salam"); 16
```

Ahay chi shod Ahay chi shod!!!

```
String newS = input.replace("Salam", "chi shod");
```

```
String sth = newS + "che khabar? " + i ;
```

```
System.out.println(sth); Ahay chi shod Ahay chi shod!!!che khabar? 5
```

- توضیح بیشتر درباره متدها و اشیاء و برنامه‌نویسی شیء‌گرا: بعداً!



- charAt
- concat مانند عملگر + کار می کند
- startsWith
- endsWith
- indexOf
- lastIndexOf
- replace
- substring
- length
- equals
- equalsIgnoreCase
- contains

اولین رخداد رشته موردنظر را پیدا می کند



- خروجی قطعه کد زیر چیست؟

```
String str = "Gholi";  
str.replace("li", "lam");  
System.out.println(str);
```

خروجی: Gholi

- توابعی مانند replace محتوای رشته را تغییر نمی‌دهند

- بلکه یک رشته جدید را بر می‌گردانند

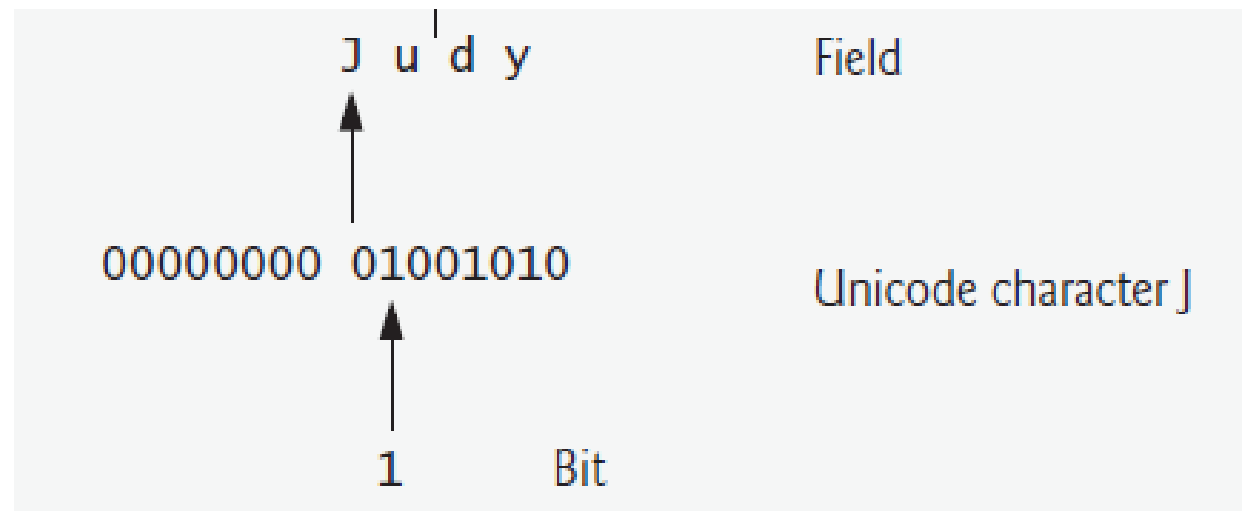
```
String str = "Gholi";  
String replaced = str.replace("li", "lam");  
System.out.println(replaced);
```

خروجی: Gholam



سلسله مراتب داده‌ها

- Bit
- Byte
- Character
- Word



- جاوا از استاندارد یونیکد پشتیبانی می کند

- **Unicode character-set**

- روش قدیمی تر: ASCII

- روش های مختلفی برای کدگذاری (**encoding**) یونیکد موجود است

- که در جاوا پشتیبانی می شوند

- مانند UTF8 ، UTF16 و UTF32

- روش هایی غیر از Unicode هم در گذشته وجود داشت

- مانند Windows-1256



کاراکترهای ویژه جاوا

- بعضی کاراکترها از کاراکترهای ویژه به شمار می‌روند
- کاراکترهای ویژه با استفاده از کاراکتر \ نمایش داده می‌شوند
- چند نمونه:

```
String s = "He said \"OK!\" and left";
```

```
He said "OK!" and left
```

• \n : خط جدید

• \t : فاصله Tab

• \" : کاراکتر \"

• \' : کاراکتر \'

• \\ : خود کاراکتر \



مثال: کاراکترهای ویژه جاوا

```
String s = "Salam!\nI am S\tA";
```

```
System.out.println(s);
```

Salam!
I am S

A

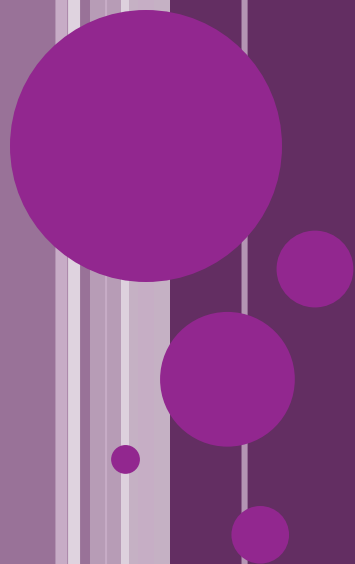
● خروجی:

```
s = "Salam\\ Ok\\ ' Bye\\";
```

```
System.out.println(s);
```

● خروجی: *Salam\ Ok ' Bye"*





Array

آرایه

آرایه (ARRAY)

- آرایه: مجموعه‌ای از داده‌های مرتبط به هم
- همه این داده‌ها، از یک نوع داده هستند
- طول آرایه ثابت است
- طول آن یک بار و در زمان ایجاد، معین می‌شود
- یک آرایه مثل مجموعه‌ای از متغیرها است
- که هر متغیر در آن یک عنصر (**element**) نامیده می‌شود

20	33	50	66	17	-3	100	25
0	1	2	3	4	5	6	7



- همه مقادیر موجود در آرایه، یک نوع واحد دارند
- مثلاً همه `int` هستند
- شماره ترتیبی هر عنصر در آرایه، اندیس (**index**) نامیده می‌شود
- عناصر آرایه به همان ترتیب در حافظه جای داده می‌شوند
- کل آرایه، یک نام دارد



Name of array (c) →	c[0]	-45
	c[1]	6
	c[2]	0
	c[3]	72
	c[4]	1543
	c[5]	-89
	c[6]	0
	c[7]	62
	c[8]	-3
	c[9]	1
Index (or subscript) of the element in array c	c[10]	6453
	c[11]	78



- ایجاد یک آرایه با ۱۰ عنصر از نوع عدد صحیح

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

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

- هر دو تعریف فوق صحیح است، ولی تعریف اول رایج تر و بهتر است

- ایجاد آرایه ای با n عنصر کاراکتری

```
char[] characters = new char[n];
```

- تغییر مقدار عنصر پنجم

```
array[5] = 12;
```

- استفاده (بازیابی) از مقدار عنصر m ام

```
char ch = array[m];
```



```
Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

double numbers[] = new double[n];

for(int i=0;i<n;i++){

    numbers[i] = scanner.nextDouble();

}

for(int i=0;i<numbers.length;i++){

    double d = numbers[i];

    System.out.println(d);

}
```



راه کوتاه ایجاد آرایه

```
char[] array = new char[3];
```

```
array[0] = 'a';
```

```
array[1] = 's';
```

```
array[2] = 't';
```

- قطعه کد فوق، می‌تواند بدین صورت بازنویسی شود:

```
char[] array = {'a', 's', 't'};
```

- چند مثال دیگر:

```
int[] numbers = {1,2,3,5,9,123};
```

```
boolean[] b = {true, true, false, true};
```



آرایه‌های چند بعدی

```
int[][] matrix = new int[3][4];
```

```
matrix[2][3] = 2;
```

```
System.out.println(matrix[2][1]);
```

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Diagram illustrating the 2D array structure and indexing:

- Column index: Points to the second index in the array notation (e.g., 1 in a[2][1]).
- Row index: Points to the first index in the array notation (e.g., 2 in a[2][1]).
- Array name: Points to the variable 'a' in the notation.



آرایه‌های چند بعدی نامتوازن

```
int[][] matrix = new int[3][];  
  
matrix[0] = new int[2];  
  
matrix[1] = new int [5];  
  
matrix[2] = new int [4];  
  
matrix[2][3] = 2;  
  
System.out.println(matrix[2][1]);
```

```
matrix[0][3] = 2; //Runtime Error
```

ArrayIndexOutOfBoundsException



ارسال آرایه به متد (PARAMETER PASSING)

```
public static void main(String[] args) {  
    int[] array = {1,2,-4,0};  
    System.out.println(max(array));  
}
```

```
static int max(int[] numbers) {  
    if(numbers == null || numbers.length == 0)  
        return -1;  
    int max = numbers[0];  
    for (int i = 1; i < numbers.length; i++)  
        if(max < numbers[i])  
            max = numbers[i];  
    return max;  
}
```



پارامترهای آرایه چند بعدی

```
int determinant(int[][] matrix) {...}  
  
    int [][] matrix = { {1,2}, {3,4}} ;  
  
    int de = determinant(matrix) ;
```

```
void check(int[][] array) {...}  
  
    int [][] unbalanced = { {1,2}, {3,4,5,6,7,8}} ;  
  
    check(unbalanced) ;
```

```
boolean f(double[][][] cube) {...}
```



نحوه ارسال آرایه به متد

- آیا محتوای آرایه هنگام ارسال به متد کپی می‌شود؟
- خیر.
- اگر متدی دارای پارامتری از نوع آرایه باشد
- عناصر آرایه در فراخوانی‌های متد کپی نمی‌شوند
- یک ارجاع به آرایه، به متد فرستاده می‌شود
- در آینده بیشتر در این باره صحبت می‌کنیم

```
int max(int[] numbers) { ... }
```



The left side of the slide features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several circles of different sizes, also in shades of purple, creating a modern, abstract design.

کوییز

- کدام یک از خطوط برنامه زیر اشکال نحوی دارد؟ به عبارت دیگر، کدام خط از این برنامه، خطای کامپایل (Syntax Error یا Compile Error) دارد؟

```
public class Temp {  
    public static void main(String[] args) {  
        //salam  
        int intValue = 2;  
        float floatValue = 2;  
        intValue = floatValue;  
        floatValue= intValue;  
        intValue = (int) floatValue;  
    }  
}
```



• خروجی قطعه برنامه زیر چیست؟

```
String name = "alireza";  
System.out.println(  
    name.indexOf('a')+name.LastIndexOf("a")+name.Length());
```

13



• خروجی قطعه برنامه زیر چیست؟

```
String[] array = new String[2];  
array[0]="a\t\n";  
array[1]="atn";  
int sum=0;  
for (int i = 0; i < array.length; i++) {  
    sum += array[i].length();  
}  
System.out.println(sum);
```

6



The left side of the slide features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several circles of different sizes, also in shades of purple, creating a modern, abstract design.

تمرین عملی

تمرین عملی، برنامه‌ای بنویسیم که:

- مقدار n را از کنسول بخواند و یک آرایه شامل n رشته از کنسول بخواند
- یک عدد از کنسول بخواند:
- اگر این عدد یک بود، میانگین طول رشته‌ها را در یک متغیر ذخیره و چاپ کند
- اگر این عدد دو بود، مجموع طول رشته‌ها را در یک متغیر ذخیره و چاپ کند
- وگرنه فقط یک پیغام بدهد
- این فرایند تکرار شود، مگر این که یکی از رشته‌ها شامل عبارت `exit` باشد
- استفاده از کامنت و تأکید روی `indentation`



The left side of the slide features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several solid purple circles of different sizes, arranged in a cluster.

تمرین

تمرین (آرایه)

1. متدی بنویسید که عناصر یک آرایه از نوع عدد صحیح را مرتب کند.
2. متدی بنویسید که دو آرایه از اعداد را با یکدیگر مقایسه کند.
 - در صورت یکسان بودن عناصر آرایه‌ها مقدار `true` و در غیر این صورت مقدار `false` را برگرداند.
3. متدی بنویسید که دترمینان یک ماتریس را برگرداند.
 - ماتریس آرایه‌ای دوبعدی است که به عنوان پارامتر به متد فرستاده می‌شود.



تمرین (رشته)

1. متدی بنویسید که نام افراد را (به صورت آرایه‌ای از رشته‌ها) به عنوان

پارامتر بگیرد و تعداد «سید»ها را برگرداند

- (تعداد کسانی که اسمشان با seyzed شروع می‌شود)

2. متدی بنویسید که مشخص کند پارامترش یک رشته «از دو سر

مساوی» است یا خیر

- یعنی رشته‌ای که اگر از انتها بخوانیمش، با خودش مساوی شود

- مثلاً radar یا hannah



The left side of the slide features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several circles of different sizes, also in shades of purple, creating a modern, abstract design.

جمع بندی

مرور و جمع‌بندی

- نحوه خواندن ورودی از کنسول
- ساختارها و دستوراتی مانند `continue` ، `break` ، `switch`
- مفهوم توضیح (`comment`) در میان کد
- مفهوم دندانه‌گذاری کد
- رشته
- کاراکترهای خاص (مانند `\n` یا `\t`)
- آرایه



● فصل‌های ۵ و ۷ از کتاب

Java How to Program (Deitel & Deitel)

5. Control Statements: Part 2

7. Arrays and ArrayLists

● تمرینهای فصل ۵



جستجو کنید و بخوانید

• کلمات و عبارات پیشنهادی برای جستجو:

- Unicode
- Character set
- Encoding
- Parameter Passing in Java
 - Call by value
 - Call by reference
- java.io.Console
- Java String Methods



A decorative graphic on the left side of the slide. It features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several circles of different sizes, also in shades of purple. The word 'پایان' is written in white Persian script to the right of the circles.

پایان



تاریخچه تغییرات

تاریخچه تغییرات

نسخه	تاریخ	توضیح
۱.۰.۰	۱۳۹۴/۱/۱۵	نسخه اولیه ارائه آماده شد و روی سایت قرار گرفت
۱.۰.۱	۱۳۹۴/۱/۱۷	بهبود تم صفحات، فوتر اضافه شد

