

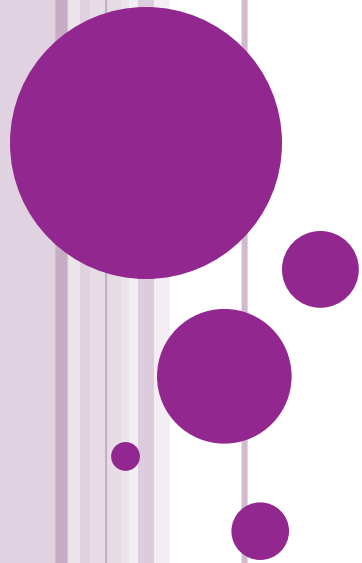
انجمن جاوا کا پتہ دیم می کند

دوره برنامه نویسی جاوا

مفاهیم اولیه زبان جاوا

JAVA BASIC CONCEPTS

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

---

- محیط‌های توسعه نرم‌افزار
- متغیرها
- انواع داده اولیه
- عملگرها
- شروط
- حلقه‌ها
- متدها



درباره محیط‌های توسعه نرم‌افزار

INTEGRATED DEVELOPMENT ENVIRONMENTS  
(IDE)

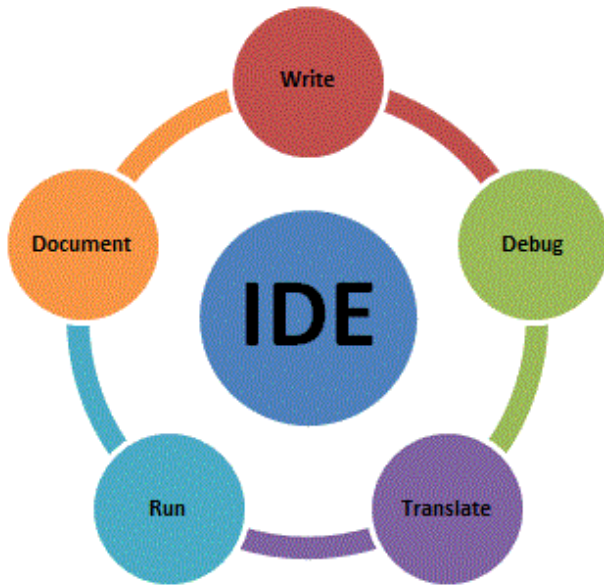
- محیط توسعه یکپارچه

## Integrated Development Environment (IDE)

- محیط‌هایی که مجموعه‌ای از امکانات مورد نیاز در برنامه‌نویسی را فراهم می‌کنند

- به صورت یکپارچه

- مثل یک جعبه‌ابزار که در دست برنامه‌نویس قرار می‌گیرد



# محیط‌های توسعه در حوزه جاوا



• محیط‌های محبوب

• Eclipse

• NetBeans

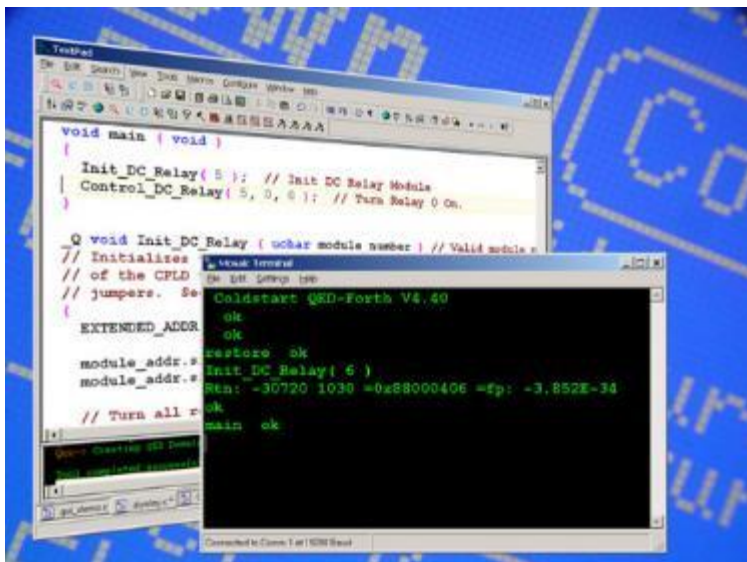
• IntelliJ IDEA

• مقایسه با .NET

• Visual Studio .NET

# مزایای محیط‌های توسعه

- تسهیل و تسريع برنامه‌نویسی
- کم کردن خطاهای برنامه‌نویس
- خودکار کردن کارهای قابل خودکارسازی
- عدم استفاده از IDE ممکن است
- مثل استفاده از notepad و cmd
- ولی برنامه‌نویسی را بسیار سخت می‌کند



# امکانات محیط‌های توسعه

- کامپایل
- اجرا
- رفع اشکال (Debug)
- نمایش مناسب خطاها و هشدارها
- کمک کردن در تولید کد
  - به خصوص تکمیل کد
- امکانات مناسب برای جستجو و تغییر در متن‌ها، کلاس‌ها و پروژه‌ها
- امکانات جانبی و فرعی
  - اتصال به مخزن کد
- امکانات بصری (Visual)



- یکی از پرستفاده‌ترین محیط‌های توسعه در جاوا

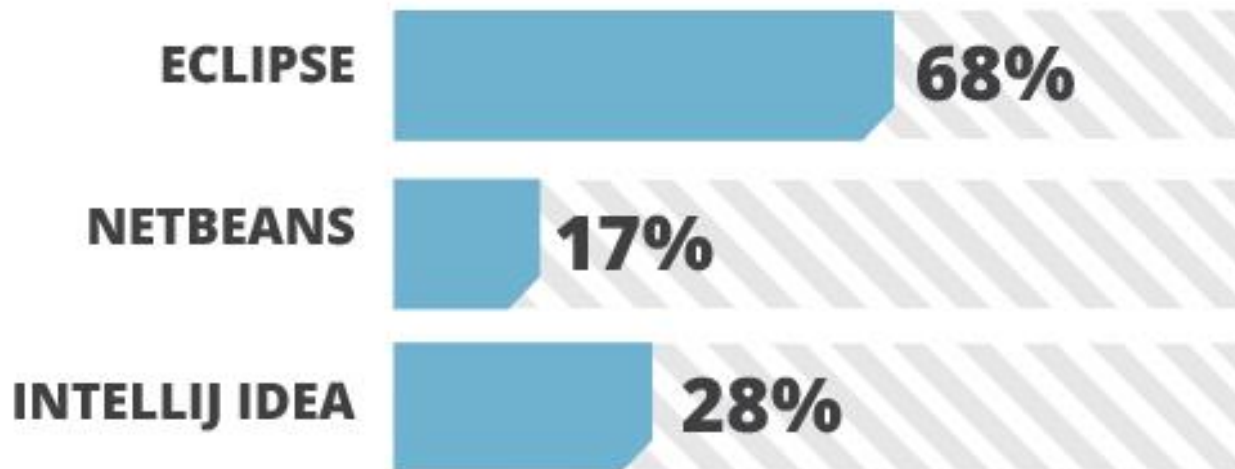
- مزایا

- رایگان

- متن‌باز

- انجمن قوی

- افزونه‌پذیری و افزونه‌های مختلف (Plugins)



- البته NetBeans و IDEA هم محیط‌های بسیار خوبی هستند

# نصب Eclipse

The screenshot shows the Eclipse Downloads page. A yellow box highlights the address bar with the URL `https://www.eclipse.org/downloads/`. Another yellow box highlights the 'Windows' dropdown menu in the header 'Eclipse Luna SR1a (4.4.1) Release for Windows'. A third yellow box highlights the 'Eclipse IDE for Java Developers' package, which is 54 MB and has been downloaded 1,090,304 times. Below this, a box highlights the 'Windows 32 Bit' and 'Windows 64 Bit' download links. At the bottom, a fourth yellow box highlights the 'Eclipse IDE for Java EE Developers' package, which is 254 MB and has been downloaded 559,298 times. The page also features a security release notice and a 'Package Solutions' section.

← → ↻ `https://www.eclipse.org/downloads/` ☆ Create account Log in

**eclipse** ≡

HOME / DOWNLOADS

» Packages Developer Builds Java™ 8 Support

**Security release:** A vulnerability with **JGit client (CVE-2014-9390)** was found, please **check for updates** or download an updated package below.

Eclipse Luna SR1a (4.4.1) Release for **Windows**

**Eclipse IDE for Java Developers**, 54 MB  
Downloaded 1,090,304 Times

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

**Windows 32 Bit**  
**Windows 64 Bit**

Package Solutions **Filter Packages ▼**

**Eclipse IDE for Java EE Developers**, 254 MB  
Downloaded 559,298 Times

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSP, Maven...



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

# کار با Eclipse

- مشاهده محل نصب Eclipse
- اجرای Eclipse
- مفهوم workspace
- ایجاد پروژه
- ایجاد کلاس
- اجرای اولین کلاس
- (فعلاً) عدم استفاده از امکانات کمکی IDE
  - مثل تکمیل کد، کلیدهای میان‌بر و ...





## متغیرها و انواع داده اولیه

# متغیر (Variable)

- بخشی از حافظه
- داده‌ای را نگه می‌دارد
- مثلاً: یک عدد صحیح، یک عدد اعشاری، یا یک رشته از حروف
- مثال:

```
int number1 = 45;
```

number1

45

number2

72

- هر متغیر نام، نوع و مقدار دارد
- نگاشت نام متغیر با محل آن در حافظه



# انواع اولیه متغیرها (Primitive Data Types)

- نوع متغیر محدوده مقادیر آن را مشخص می‌کند
- مثال:
- متغیری از نوع `int`، عددی صحیح بین  $-2^{31}$  تا  $2^{31}-1$  را نگه می‌دارد
- نکته مهم:
- برخلاف بسیاری از زبان‌های دیگر (مثل `C++`)،
- در زبان جاوا، بازه مقادیر ممکن در این متغیرها کاملاً مشخص است
- این بازه، وابسته به محیط (کامپایلر، سیستم‌عامل، سخت‌افزار و ...) نیست
- این تمهیدی برای مستقل از سکو بودن جاوا است
- Platform Independence



# انواع داده اولیه (Primitive Data Types)

• اعداد صحیح

• byte, short, int, long

• کاراکتر

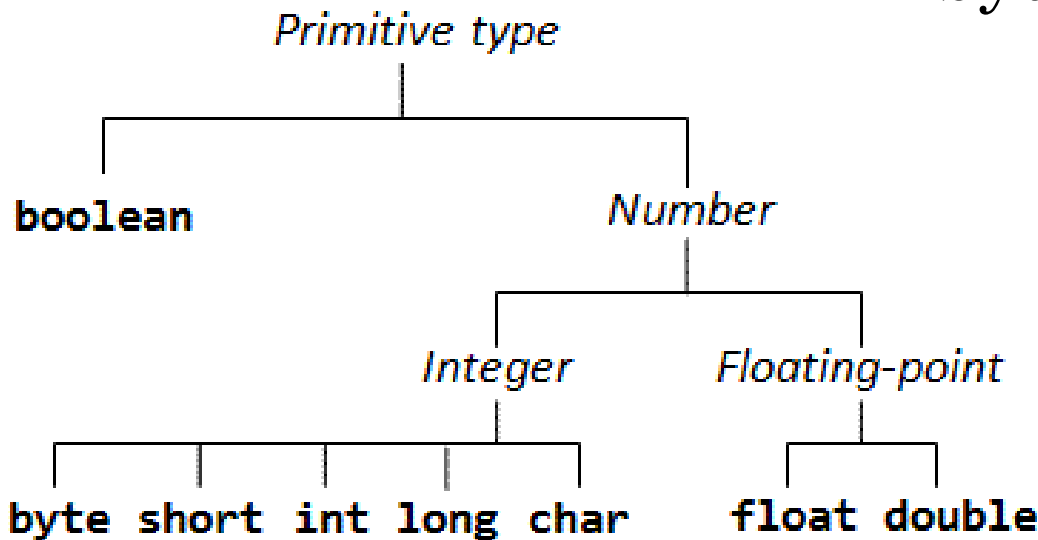
• char

• اعداد اعشاری

• float, double

• مقدار منطقی (درست/غلط)

• boolean





# انواع داده اولیه

Primitive type	Size	Minimum	Maximum
boolean	—	—	—
char	16 bits	Unicode 0	Unicode $2^{16}-1$
byte	8 bits	-128	+127
short	16 bits	$-2^{15}$	$+2^{15}-1$
int	32 bits	$-2^{31}$	$+2^{31}-1$
long	64 bits	$-2^{63}$	$+2^{63}-1$
float	32 bits	IEEE754	IEEE754
double	64 bits	IEEE754	IEEE754



# مقادیر ثابت (Literals)

```
byte a = 12;  
short b = 1232;  
int c = 190823;  
long l = 1233;  
l = 293847298347L;  
  
boolean e = true;  
e = false;
```

```
char ch = 'a';  
ch = '4';  
  
double d = 3.14;  
d = 4763.129837D;  
  
float f = 12.23F;
```



# عملگرها (Operators)

- عملی بر روی مقادیر و متغیرها انجام می‌دهند

- مثال:

- عملگر مقداردهی (assignment)

```
number = 3 ;
```

- عملگر جمع یا ضرب

```
a = b + 12 ;
```

```
pi = 3.14 ;
```

```
circle = 2 * radius * pi ;
```

# عملگرهای ریاضی

Java operation	Operator	Algebraic expression	Java expression
Addition	+	$f + 7$	<code>f + 7</code>
Subtraction	-	$p - c$	<code>p - c</code>
Multiplication	*	$bm$	<code>b * m</code>
Division	/	$x / y$ or $\frac{x}{y}$ or $x \div y$	<code>x / y</code>
Remainder	%	$r \bmod s$	<code>r % s</code>

`a = b / c ;`

● در مورد عملگر تقسیم

● اگر هر دو عملوند، از نوع غیراعشاری باشند

● خروجی، خارج قسمت تقسیم (بدون اعشار) خواهد بود

`System.out.println(3/2);` 1

`System.out.println(3.0/2);`

`System.out.println(3/2.0);`

1.5

`System.out.println(3.0/2.0);`

```
double b = 3.0;
int c = 2;
double a = b/c;
```

a=1.5

```
int b = 3;
int c = 2;
float a = b/c;
```

a=1.0

```
short b = 3;
long c = 2;
float a = b/c;
```



# اولویت عملگرها (Operator Precedence)

Operator(s)	Operation(s)
*	Multiplication
/	Division
%	Remainder
+	Addition
-	Subtraction
=	Assignment

- چند عملگر از یک دسته: از چپ به راست اجرا می شوند



## مثال (اولویت عملگرها)

- مقدار متغیر number چه خواهد بود؟

```
int number = 2 + 3 * 4 / 2 ;
```

- پاسخ صحیح: ۸





























- البته همواره پرانتزگذاری برای پرهیز از ابهام توصیه می‌شود:

```
int number = 2 + ((3 * 4) / 2) ;
```

- مثال دیگر: `z = p * r % q + w / x - y;`

عملگرها با چه ترتیبی اجرا می‌شوند؟

## Equality operators

$$x == y$$
$$\neq \quad \quad \quad ! =$$
$$x \neq y$$

## Relational operators

$$\frac{1}{2} \leq \frac{1}{2} \leq \frac{1}{2}$$
 $x > y$  $\angle$  $x < y$ 
$$M \geq$$
$$x \geq y$$
$$\mathbb{V} \leq \mathbb{V}^{\text{max}}$$
$$x \leq y$$



# شرکت پذیری عملگرها (Associativity)

- وقتی دو عملگر با یک اولویت یکسان در یک عبارت قرار می گیرند،
- ارزیابی این عبارت با توجه به خاصیت «شرکت پذیری» عملگرها صورت می پذیرد
- شرکت پذیری: «راست به چپ» یا «چپ به راست»
- مثال:  $x = y = z = 17$  به صورت  $x = (y = (z = 17))$  اجرا می شود
- زیرا شرکت پذیری عملگر مقداردهی، راست به چپ است
- مثال:  $72 / 2 / 3$  به صورت  $(72 / 2) / 3$  اجرا می شود
- زیرا شرکت پذیری عملگر تقسیم، چپ به راست است



# عملگرهای تک عملوندی (unary Operators)

```
int a = 12;
```

```
a++;
```

```
a--;
```

```
++a;
```

```
--a;
```

```
int b = --a;
```

```
b = a--;
```

```
b = a++;
```

```
b = ++a;
```

```
b = -a;
```

```
a = +2;
```

```
b = ++a - -a;
```

```
boolean t = true;
```

```
boolean f = false;
```

```
t = !f;
```

```
t = ! (a==b) ;
```

```
f = t;
```

# عملگرهای منطقی (Logical Operators)

---

```
boolean x = true, y = false;
```

```
boolean z = x && y;
```

```
z = x || y;
```

```
z = x && true || y && false;
```

```
z = !x;
```

# عملگرهای ترکیبی

- عملگرهای مختلف را می‌توانیم با عملگر مقداردهی ترکیب کنیم
- اگر اولین عملوند در سمت راست مقداردهی همان عملوند سمت چپ باشد
- عملگرهای حاصل، خلاصه دو عملگر هستند
- مثال:

$a = a/10; \rightarrow a/=10;$

$b = b*10; \rightarrow b*=10;$

$a = a + 1; \rightarrow a+=1$  ; یا  $a++;$



# عملگر سه عملوندی شرطی

● نحوه استفاده:

```
TYPE value = CONDITION ? Val_True : Val_False;
```

● مثال:

```
double area = 2 * 2 * 3.14;
```

```
int value = area > 10 ? 1 : -1;
```

```
//value = 1;
```

```
if(area>10)
    value = 1;
else
    value = -1;
```

# خلاصه عملگرها

Operator	Description	Associativity
++ --	unary postfix increment unary postfix decrement	right to left
++ -- + - !	unary prefix increment unary prefix decrement unary plus unary minus unary logical negation	right to left
* / %	multiplication division remainder	left to right
+ -	addition or string concatenation subtraction	left to right



## خلاصه عملگرها (۲)

<	less than	left to right
<=	less than or equal to	
>	greater than	
>=	greater than or equal to	
==	is equal to	left to right
!=	is not equal to	
&&	conditional AND	left to right
	conditional OR	left to right
?:	conditional	right to left
=	assignment	right to left
+=	addition assignment	
-=	subtraction assignment	
*=	multiplication assignment	
/=	division assignment	
%=	remainder assignment	



- استفاده از `println` و `print` برای چاپ خروجی

- `System.out.println(a);`
- `System.out.print(x);`
- `System.out.println(2+5*a);`

- یک مقدار یا متغیر را در خروجی چاپ می کند

- کنسول (console)

- خروجی استاندارد (standard output)

- یادآوری: اولین برنامه که رشته `Salam` را چاپ می کرد:

- `System.out.println("salam!");`



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

# کوئیز ۱: خروجی این برنامه چیست؟

```
int a;
```

```
a = 12;
```

```
a+= 2;
```

```
int b;
```

```
b = 4;
```

```
b++;
```

```
b = a*b;
```

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

## کوئیز ۲: خروجی این قطعه برنامه چیست؟

```
double x;
```

```
x = 5 / 2;
```

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

```
x = 7 + 3 * 6 / 2 - 1;
```

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

```
x = 2 % 2 + 2 * 2 - 2 / 2;
```

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

```
x = (3 * 9 * (3 + (9 * 3 / (3))));
```

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

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 متغیرها و عملگرها

- انواع متغیر
- عملگرها
- رفتار تقسیم در تقسیم اعداد صحیح
- تفاوت عملگر افزایش ( $++$ ) یا کاهش به صورت پیشوندی و پسوندی
- آشنایی با متدهای `print` و `println` برای چاپ خروجی
- `System.out.println(a);`





# کنترل جریان برنامه

- امکانی بر کنترل جریان اجرای برنامه
- اگر «فلان شرط» آن گاه «این کارها را بکن» وگرنه اگر «بههمان شرط»، «آن کار» وگرنه ...

```
if (x>y)
```

```
    System.out.println("X is greater than Y");
```

```
else if (x==y)
```

```
    System.out.println("X is equal to Y");
```

```
else
```

```
    System.out.println("Y is greater than X");
```

```
boolean condition = x>y;
```

```
if(condition)
```

```
    System.out.println("X is greater than Y");
```

```
else
```

```
    System.out.println("Y >= X");
```



# مفهوم بلوک

● فرض کنید بخواهیم:

- در صورتی که یک شرط برقرار بود، چند عملیات انجام شود
- و اگر آن شرط برقرار نبود، هیچ یک انجام نشود

**if (x>=y)**

```
{  
    a++;  
    b = x;  
    c = x/y ;  
}
```

**else**

```
{  
    c = y/x;  
    a--;  
    System.out.println("Y > X") ;  
}
```



- برخی از زبان‌ها goto را ممکن می‌سازند
- امکانی برای پرش مستقیم از یک خط برنامه به یک خط دیگر
- روشی برای کنترل جریان اجرای برنامه
- توصیه نمی‌شود، زیرا برنامه پیچیده و غیرساخت‌یافته می‌شود
- دستور goto در جاوا یک کلمه رزرو شده است (reserved word)
- اما استفاده از آن ممنون است!



The left side of the image 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.

حلقه‌ها



- امکانی برای انجام چندباره یک عملیات
- تعداد مشخصی از تکرار، یا تکرار تا زمان حصول یک شرط
- حلقه‌های جاوا:

for •

while •

do-while •



# حلقه while

```
long counter=1;
```

```
while (counter<10)
```

بلوک

```
{
```

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

```
    counter++;
```

```
}
```

- خروجی این برنامه؟

- چاپ اعداد ۱ تا ۹



# حلقه do-while

- شرط حلقه، در انتهای عملیات بررسی می‌شود
- بنابراین بدنه do-while حداقل یک بار اجرا می‌شود

```
long counter=0;  
  
do {  
  
    counter++;  
  
    System.out.println(counter);  
  
} while (counter<10);
```

- خروجی این برنامه؟
- چاپ اعداد ۱ تا ۱۰

- پرستفاده‌ترین حلقه

یک دستور (یک عملیات ; یک شرط ; یک عملیات) **for**

- مثال:

```
for (int i = 1; i <= 10; i++) {  
    System.out.println(i);  
}
```

- خروجی این برنامه؟

- چاپ اعداد ۱ تا ۱۰



# while for با کمک

```
for (X; Y; Z) {  
    body () ;  
}
```

```
X;  
while (Y) {  
    body () ;  
    Z;  
}
```



# مثال باز نویسی

```
for (int i = 1; i <= 10; i++) {  
    System.out.println(i);  
}
```

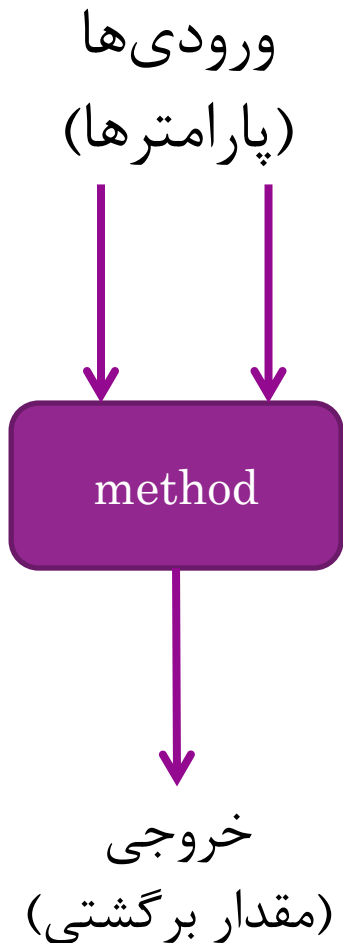
```
int i=1;  
while (i<=10) {  
    System.out.println(i);  
    i++;  
}
```





آشنایی اولیه با متدها

# متد (Method)



- یک متد مثل یک دستگاه است
  - تعدادی ورودی می‌گیرد
  - کارهایی روی ورودی‌ها انجام می‌دهد
  - خروجی تولید می‌کند
- تعداد ورودی‌ها: صفر یا بیشتر
- تعداد خروجی‌ها: صفر یا یک
- ورودی‌های متد: پارامتر (Parameter)
- خروجی متد: مقدار برگشتی (Return Value)
- نام‌های دیگر
  - تابع (Function)
  - روال یا رویه (Procedure)

مقدار  
برگشتی

نام متد

پارامترها

```
double add(double a, double b) {  
    double result = a+b;  
    return result;  
}
```

```
double x = 3;
```

```
double y = 4;
```

```
double sum = add(x,y);
```

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



- متد ممکن است بدون پارامتر باشد
- مثل متد `System.out.println`
- ممکن است بدون خروجی باشد
- نوع خروجی را **void** قرار می‌دهیم
- از دستور `return` استفاده نمی‌کنیم

```
void simple() {  
    System.out.println(": -");  
}
```



```
public class PrimeNumbers {  
    public static void main(String[] args) {  
        for(int num=0;num<20;num++)  
            if(isPrime(num))  
                System.out.println(num);  
    }  
    static boolean isPrime(int number){  
        if(number<2)  
            return false;  
        for(int i=2;i<number;i++)  
            if(number%i==0)  
                return false;  
        return true;  
    }  
}
```

- نکته: فعلاً از کلمه static قبل از تعریف متد استفاده کنید

- توضیح بیشتر درباره دلیل و معنی این کلمه: بعداً!



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

- متدی نوشتیم که مشخص می کند پارامترش عدد اول است یا خیر  
`isPrime` •
- این متد را با `while` ، `do-while` و `for` بازنویسی کنید
- متدی بنویسید که یک عدد به عنوان پارامتر بگیرد و اعداد اول کوچکتر از آن را چاپ کند  
`printPrimes` •
- برنامه‌ای بنویسید که از این متدها استفاده کند (فراخوانی)





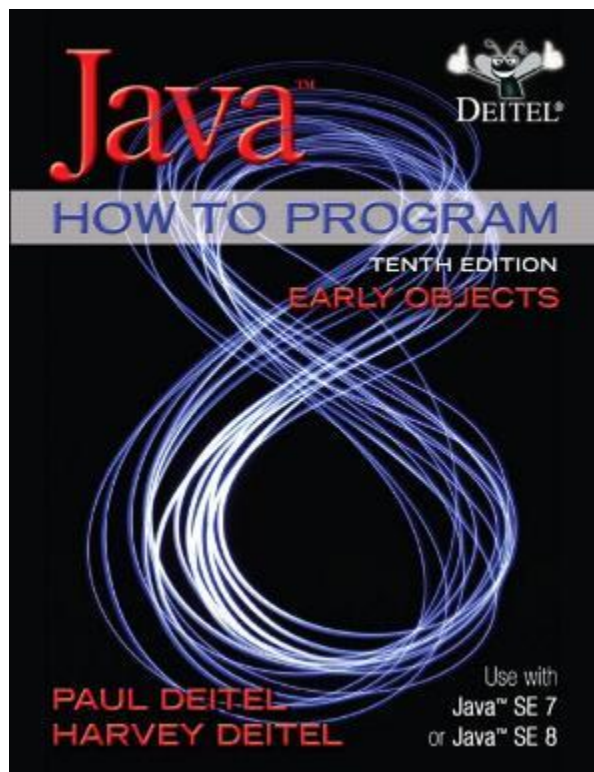
The left side of the slide features a series of vertical stripes in various shades of purple and white. Overlaid on these stripes are several circles of different sizes, also in shades of purple, creating a modern, abstract design.

جمع بندی

- کار با محیط‌های توسعه
- Eclipse
- معرفی انواع داده اولیه
- متغیرها
- آشنایی با عملگرها
- اولویت و شرکت‌پذیری
- آشنایی با متدها، شروط و حلقه‌ها

- فصل‌های اول، دوم، چهارم و پنجم کتاب دایتل

## Java How to Program (Deitel & Deitel)



- 1- Introduction
- 2- Introduction to Java Applications
- 4- Control Statements: Part 1
- 5- Control Statements: Part 2

- تمرین‌های همین فصل‌ها از کتاب دایتل



# درباره تمرین

- خواندن کتاب یا دیدن اسلاید و ویدیو، کافی نیست
- تمرین عملی بسیار مهم است
- حتی اگر فکر می کنید همه مطالب را به خوبی یاد گرفته اید
- (البته مطالعه، دانش و مهارت شما را عمیق و ذهنتان را منظم می سازد)



# تمرین‌های عملی

1. برنامه‌ای بنویسید که دنباله فیبوناچی را تا اعداد کوچکتر از ۱۰۰۰ چاپ کند

- در صورت امکان به صورت بازگشتی (recursive) پیاده‌سازی کنید

2. متدی بنویسید که مشخص کند پارامترش یک عدد «از دو سر مساوی» (palindrome) است یا خیر

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

- مثلاً `palindrome(1221)` مقدار `true` و `palindrome(122)` مقدار `false` برمی‌گرداند

3. متدی بنویسید که مشخص کند آیا پارامترش توانی از دو است؟

- مثلاً ۱ و ۲ و ۸ و ۶۴ توانی از دو هستند، ولی ۳ و ۶ نیستند



## تمرین (ادامه)

4. برنامه‌ای بنویسید که کارکترهای 'a' تا 'z' را یک‌درمیان چاپ کند

5. متدی بنویسید که پارامترش شعاع دایره است و مقدار برگشتی آن مساحت دایره.

6. تابعی بنویسید که با کمک کاراکتر ستاره (\*) یک مثلث متساوی الساقین با قاعده‌ای عمودی رسم کند.

- اندازه قاعده به عنوان پارامتر دریافت شود.
- مثلاً با پارامتر ۵، این ساختار را چاپ کند:

```
*  
***  
*****  
***  
*
```

7. متدی بنویسید که فاکتوریل پارامترش را برگرداند

- هم به صورت حلقه‌ای و هم به صورت بازگشتی پیاده‌سازی کنید



# جستجو کنید و بخوانید

• کلمات و عبارات پیشنهادی برای جستجو:

- Java Bitwise Operators
- IDE
- Eclipse
- IntelliJ IDEA
- NetBeans
- GOTO statement



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

