

به نام خدا



برنامه‌سازی ییشرفته

دانشگاه شهید بهشتی · دانشکده مهندسی کامپیوتر

دکتر مجتبی وحیدی اصل

آشنایی با جاوا

آراس ولیزاده

فهرست مطالب

1. چرا جاوا؟
2. شاخص‌های محبوبیت زبان‌های برنامه‌نویسی
3. خصوصیات زبان جاوا
4. ابزارهای لازم
5. سرفصل‌های درس
6. نکات مهم موفقیت در درس

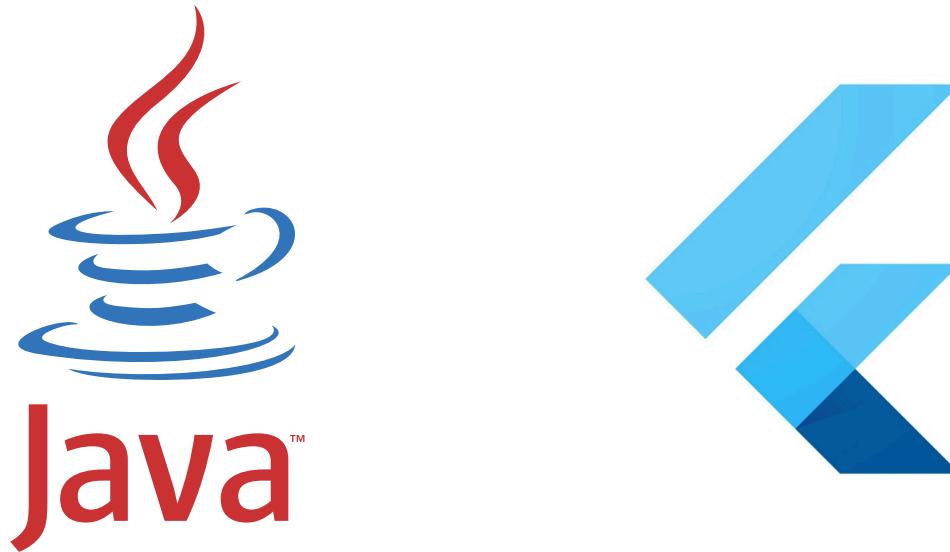
مقدمه

به درس **برنامه‌سازی پیشرفته** خوش اومدید؛ در این درس با زبان **جاوا**، یکی از پرکاربردترین زبان‌های برنامه‌نویسی جهان، آشنا می‌شویم. در این دوره، از مفاهیم پایه‌ای شروع می‌کنیم و تا مباحث پیشرفته‌ای مانند شیءگرایی، مدیریت استثناهای و برنامه‌نویسی چندریسمانی پیش خواهیم رفت. هدف ما این هستش که شما نه تنها با سینتکس و ساختار زبان جاوا آشنا بشوید، بلکه نحوه تفکر برنامه‌نویسی حرفه‌ای و حل مسئله با رویکرد شیءگرا را نیز فرا بگیرید. این مهارت‌ها پایه‌ای محکم برای ورود به دنیای توسعه نرم‌افزار و کسب تجربه در بازار کار فراهم خواهد کرد.

نکته: این دوره نیازمند تمرین مداوم و حل مسائل عملی است. پس آماده باشید تا با کدنویسی زیاد، مهارت‌های خود را تقویت کنید!

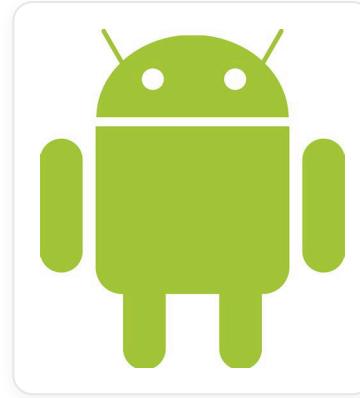
چرا جاوا

در که محبوب‌ترین متدولوژی توسعه نرم‌افزار است نیز آشنا خواهید شد (OOP) علاوه بر یادگیری ساختار و ویژگی‌های جاوا، با مفاهیم [برنامه‌نویسی شیء‌گرا](#) ادامه، نگاهی بر برنامه‌نویسی در [فریم‌ورک فلاتر](#) نیز خواهیم داشت تا بتوانید از دانش خود در توسعه اپلیکیشن‌ها استفاده کنید.



در اکوسیستم نرم‌افزار امروز، [جاوا](#) همچنان بین زبان‌های پراستفاده قرار دارد: بر اساس نظرسنجی سال ۲۰۲۰ استک‌اورفلو، حدود ۳۹٪ از توسعه‌دهندگان جاوا زبان رسمی توسعه اندروید می‌باشد. بسیاری از نیز در جمع ۴ زبان برتر است TIOBE حرفه‌ای در یک سال گذشته با جاوا کار کرده‌اند، و جاوا در شاخص از منظر بازار، هزینه مصرف‌کننده در به جاوا نوشته می‌شوند و میلیون‌ها برنامه‌نویس جاوا در این حوزه فعال هستند client-server نرم‌افزارهای تحت وب اپ‌های موبایل در سال ۲۰۲۰ حدود ۱۵۰ میلیارد دلار بوده (۱۳٪ رشد نسبت به ۲۰۲۳) که نشان می‌دهد فرصت اقتصادی اکوسیستم اپ بسیار بزرگ است؛ بخش در سمت سرور، جاوا همچنان ستون فقرات بسیاری از سامانه‌های وب. قابل توجهی از این اپ‌ها به صورت بومی یا هیبرید برای اندروید توسعه می‌یابند

و ابزار بالغ، برای سرویس‌های در مقیاس سازمانی انتخاب رایجی Spring (به‌ویژه با اکوسیستم) و بهدلیل پایداری، و enterprise-سرور است محسوب می‌شود.



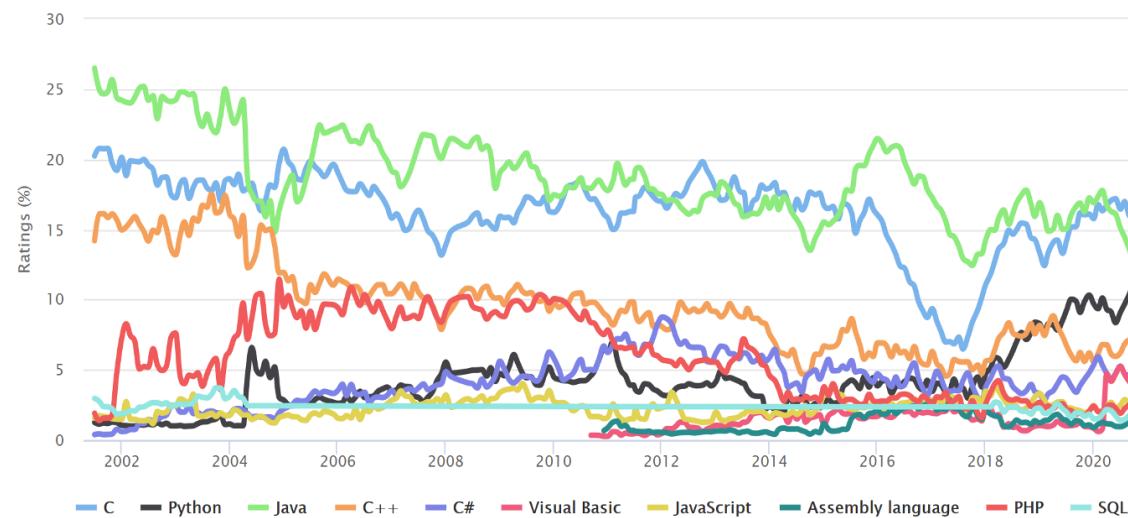
چرا جاوا محبوب است؟

محبوبیت یک زبان برنامه‌نویسی را می‌توان از چند شاخص مهم سنجید:

- تعداد آگهی‌های شغلی که به دنبال برنامه‌نویس مسلط به آن زبان هستند
- میانگین حقوق و مزایای پرداختی به توسعه‌دهندگان آن زبان
- میزان فروش و تنوع کتاب‌ها و منابع آموزشی مرتبط
- تعداد کدها و پروژه‌های متن‌باز و نمونه‌کدهای منتشرشده در اینترنت
- حضور فعال در جوامع و انجمن‌های برنامه‌نویسی

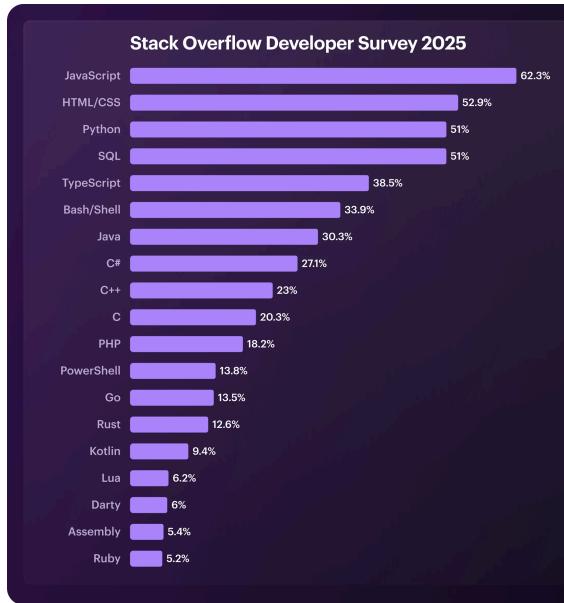
TIOBE Programming Community Index

Source: www.tiobe.com

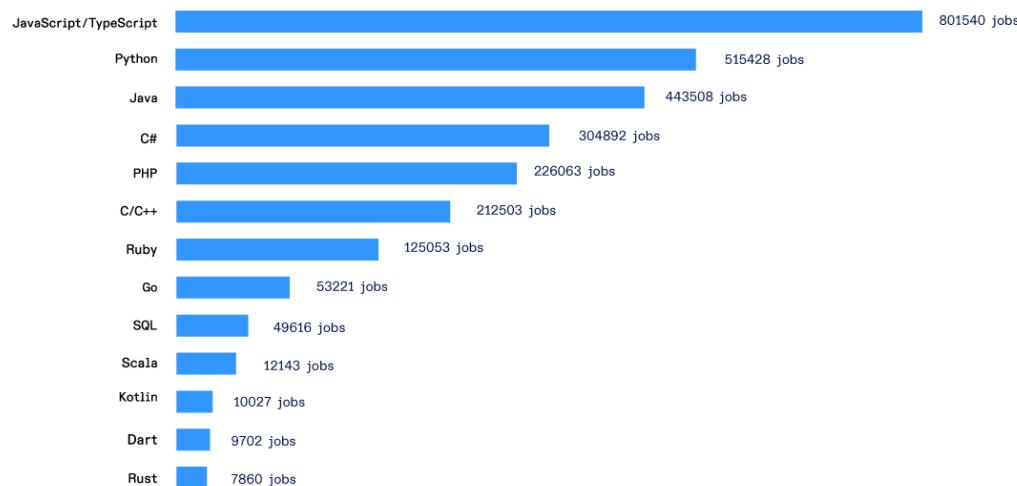


The TIOBE Ranking 2025

FEB 2025	FEB 2024	LANGUAGE	RATINGS	CHANGE
1	1	 Python	23.88%	+8.72%
▲ 2	3	 C++	11.37%	+0.84%
▲ 3	4	 Java	10.66%	+0.79%
▼ 4	2	 C	9.84%	-1.14%
5	5	 C#	4.12%	-3.41%



Most Demanded Programming Languages by Number of Jobs

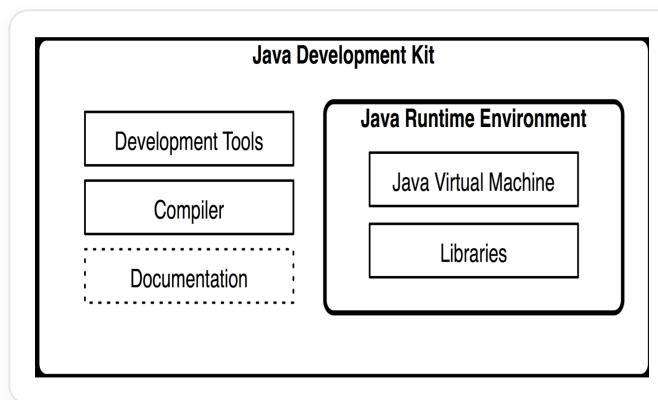


Java Development Kit (JDK)

برای نوشتن، کامپایل و اجرای برنامه‌های جاوا به **JRE** بتهنایی فقط اجرای برنامه را ممکن می‌کند.

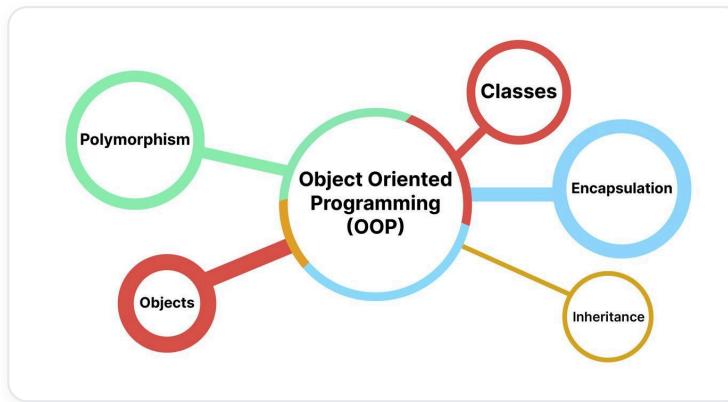
- ابزارهای توسعه مانند `javac` (کامپایل)، `java` (اجرای دیباگر و ابزار پکیج/مستندسازی).
 - تبدیل کد منبع `.java` به بایت کد `.class` قابل اجرا روی JVM.
 - مستندات زبان و API‌ها برای توسعه سریع‌تر و درست‌تر.
 - شامل: **JRE** برای اجرای بایت‌کد.
 - **JVM** کتابخانه‌های استاندارد برای ورودی/خروجی، شبکه، کالکشن‌ها و ...
 - **Libraries**

جمع‌بندی: **JDK = ابزارهای توسعه + JRE**، در حالی که **JRE = فقط اجرای برنامه**.

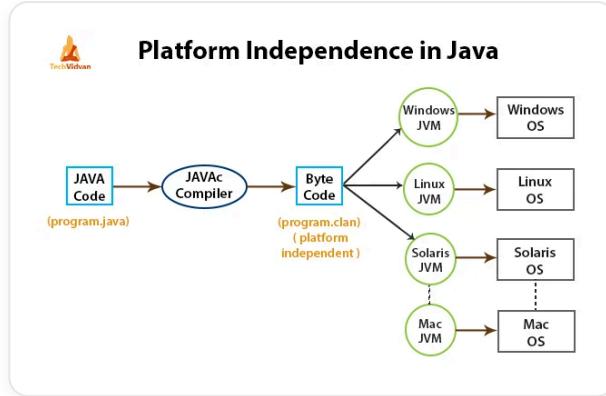


خصوصیات زبان جاوا

- **سادگی:** سینتکس آشنا شبیه C اما بدون اشاره‌گر؛ مدیریت حافظه خودکار با Garbage Collection.
- **شیءگرایی (Object-Oriented):** مدل‌سازی دنیای واقعی با کلاس/شیء، تعامل بهتر اجزا، و استفاده مجدد از کد.



- **قابل حمل و مستقل از معماری:** یک کد جاوا روی سامانه‌های مختلف بدون تغییر اجرا می‌شود.
- **ترکیب کامپایل و تفسیر:** در جاوا ابتدا کد منبع توسط کامپایلر به قالبی به نام Bytecode تبدیل می‌شود. این کد میانی مستقل از سیستم‌عامل است و توسط ماشین‌مجازی جاوا (JVM) در زمان اجرا تفسیر یا به کد بومی ترجمه می‌شود. این فرآیند باعث می‌شود برنامه بدون تغییر روی سیستم‌عامل‌ها و معماری‌های مختلف اجرا شود (Platform-Independent).



- **چندریسمانی (Multithreading):** قابلیتی که به برنامه اجازه می‌دهد چند بخش (Thread) را به‌طور همزمان اجرا کند. این ویژگی باعث می‌شود وظایف سنگین مانند پردازش داده، رندر گرافیک، یا ارتباطات شبکه‌ای بدون ایجاد توقف در عملکرد اصلی برنامه انجام شوند. جاوا ابزارها و کلاس‌های داخلی متنوعی برای مدیریت Thread‌ها، همگام‌سازی (Synchronization) و جلوگیری از تداخل داده‌ها فراهم می‌کند، که در نتیجه کارایی و پاسخ‌گویی برنامه بهبود می‌یابد.
- **امن و قابل اتکا (Secure & Robust):** مدل امنیتی قوی، بررسی خطا در زمان کامپایل/اجرا، عدم دسترسی مستقیم به حافظه.

WORA (Write Once, Run Anywhere) اصل

یکی از مهم‌ترین مزیت‌های جاوا اصل **یک بار بنویس، همه‌جا اجرا کن** است. این یعنی شما یک بار کد را می‌نویسید و پس از کامپایل به فایل `class`، این کد می‌تواند روی هر سیستم‌عاملی که **JRE** نصب دارد اجرا شود، بدون اینکه نیاز به تغییر در کد منبع باشد.

1. نوشتن کد منبع جاوا (Java Source)
2. کامپایل کد منبع توسط کامپایلر جاوا و تولید Bytecode
3. اجرای Bytecode بر روی هر محیطی که JRE دارد (Windows, Linux/Unix, macOS, Android)

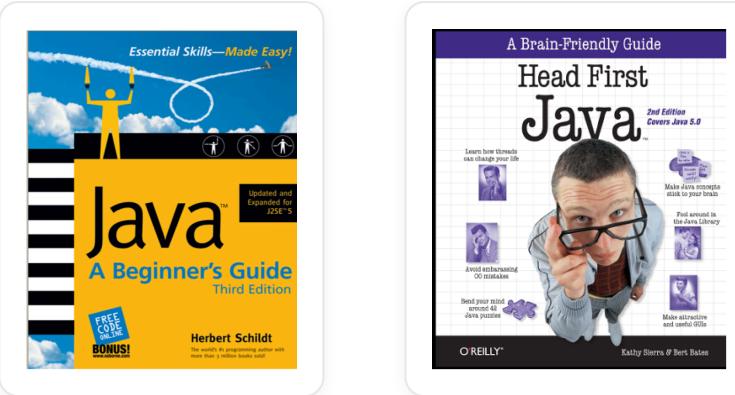
نتیجه این ویژگی، کاهش هزینه توسعه و نگهداری، و امکان توسعه آسان نرم افزار برای چندین پلتفرم به صورت همزمان است.



منابع درس

برای یادگیری بهتر زبان جاوا، منابع متنوعی در دسترس هستند که هر کدام با رویکردی متفاوت به آموزش این زبان پرداخته‌اند.

- نوشته Herbert Schildt، مناسب برای شروع یادگیری از پایه.
- نوشته Harvey Deitel و Paul Deitel، با مثال‌ها و تمرین‌های فراوان.
- نوشته Bert Bates و Kathy Sierra، رویکرد تصویری و مفهومی برای یادگیری آسان‌تر.
- نوشته Bruce Eckel، مناسب برای درک عمیق مفاهیم شیء‌گرایی و ساختار جاوا.



اولین برنامه به زبان جاوا

در این بخش، قصد داریم اولین برنامه خود را به زبان جاوا بنویسیم. برای این کار از [Jupyter Notebook](#) استفاده می‌کنیم. [Jupyter](#) یک محیط تعاملی و قدرتمند برای نوشتمن و اجرای کد، تحلیل داده و مستندسازی است که به طور پیش‌فرض با کرنل Python اجرا می‌شود. چون قصد داریم کدهای جاوا را اجرا کنیم، لازم است یک کرنل جاوا (مثل [JBang](#)) نصب کنیم تا Jupyter بتواند کدهای جاوا را پردازش و اجرا کند. این کار به ما امکان می‌دهد تا در همان محیط تعاملی Jupyter، کد جاوا بنویسیم، اجرا کنیم و خروجی آن را بلافاصله ببینیم. ابتدا cell زیر را با استفاده از کرنل پایتون اجرا کنید و پس از نصب، می‌توانید کرنل جاوا را انتخاب کنید.

```
In [ ]: !pip install jbang
import jbang
jbang.exec("trust add https://github.com/jupyter-java")
jbang.exec("install-kernel@jupyter-java")
```

```
In [ ]: public class Welcome3 {
    public static void main(String[] args) {
        System.out.println("Welcome\ninto\nJava\nProgramming!");
    }
}
```

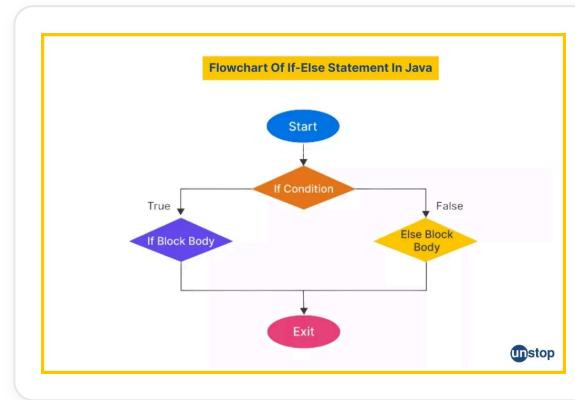
```
}

// To print the result:
Welcome3.main(new String[]{});
```

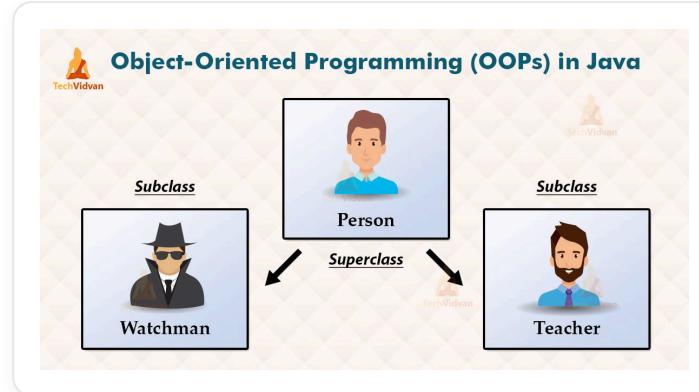
سرفصل‌های آموزشی دوره جاوا

در این دوره، به صورت گام‌به‌گام مفاهیم کلیدی و پیشرفته زبان جاوا آموزش داده خواهد شد. مباحث شامل:

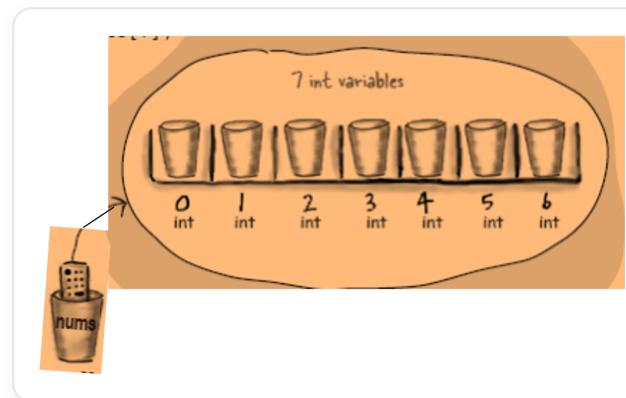
- **ساختارهای کنترلی:** شامل if ساده، if..else، do..while، for، while و دستورات break و continue.



- **مفاهیم شیء‌گرایی پیشرفته:** آشنایی با ارث بری، پلی‌مورفیسم و نحوه استفاده مؤثر از کلاس‌ها، متدها و متغیرها.



- آرایه‌ها و رشته‌ها: کار با آرایه‌های یکبعدی و چندبعدی، ایجاد، تغییر و حذف عناصر و ارسال آرایه‌ها به متدها.

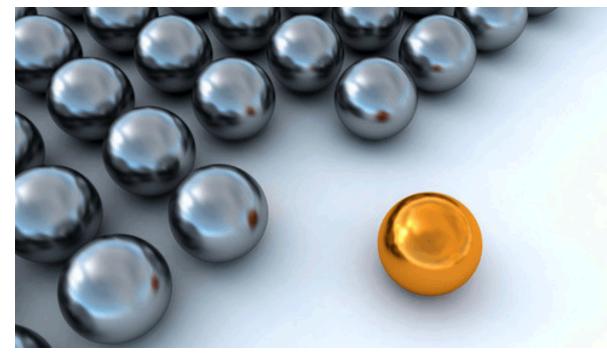


- متدها و مازول‌ها: متدهای ایستا و فیلد های ایستا، تعریف متدهای چندپارامتری، آشنایی با بسته‌های API جاوا، سربارگذاری و بازنویسی متدها.

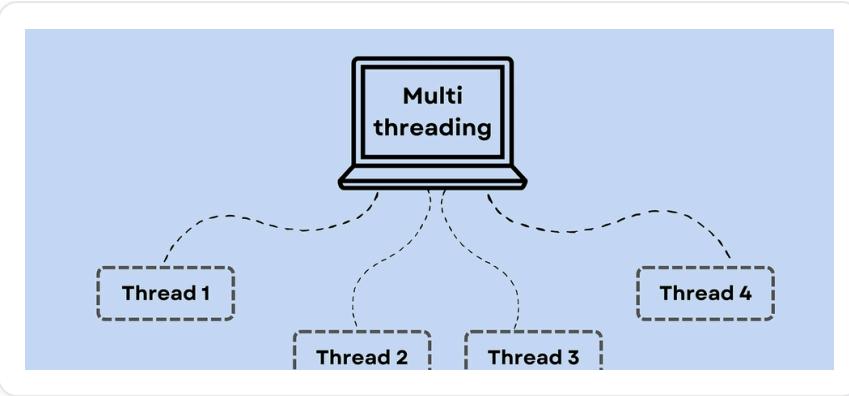
return type method name value passed to the method

```
int total( int aNumber) {  
  
    int a_Value = aNumber + 10;  
  
    return a_Value;  
}
```

- **مدیریت استثنایها:** سلسله مراتب استثنایها، استثنای زنجیره ای، تعریف استثنای جدید و نحوه کنترل خطاهای در برنامه.



- **سایر مباحث پیشرفته:** کار با فایل‌ها و ورودی/خروجی در جاوا، مقدمات برنامه‌نویسی شبکه، مفاهیم مقداردهی ایستا و پویا، برنامه‌نویسی چندریسمانی (Multithreading)، کار با مجموعه‌ها (Collections)، برنامه‌نویسی جنریک (Generic Programming) و بسیاری مطالب دیگر.



چند نکته مهم

- **اهمیت تمرین پیوسته:** تمرین مداوم یکی از کلیدی‌ترین عوامل موفقیت در یادگیری این درس است. توصیه می‌شود علاوه بر تمرین‌های ارائه شده در کلاس، به صورت مستقل نیز مسائل و مثال‌های بیشتری را حل کنید. این کار باعث تثبیت مفاهیم و افزایش مهارت شما در حل مسئله می‌شود.
- **شرط قبولی در درس:** برای پاس کردن این درس، لازم است که دانشجو حداقل ۰٪ از مجموع نمره میان‌ترم و پایان‌ترم را کسب نماید.
- **همکاری در انجام تمرین‌ها:** می‌توانید برای فهم بهتر تمرین‌ها با همکلاسی‌های خود مشورت کنید و ایده‌ها را به اشتراک بگذارید. با این حال، اجرای نهایی تمرین‌ها باید توسط خودتان انجام شود تا هم مهارت‌های فردی‌تان تقویت شود و هم اصول اخلاق آکادمیک رعایت گردد. هرگونه کپی‌برداری مستقیم بدون ذکر منبع، تخلف محسوب می‌شود.
- **ابزار‌های هوش مصنوعی در درس ما:** استفاده از ابزار‌های هوش مصنوعی برای نوشتن کد جایز نیست و در صورت مشاهده تقلب تلقی می‌شود؛ همچنین تمارین و پروژه درس توسط سیستم تقلب یاب کوئرا چک می‌شود و در صورت مشاهده درصد های بالا این مورد هم تقلب تلقی می‌شود.
- **کوییزها و ارزیابی مستمر:** در طول ترم تعدادی کوییز (توسط تیم حل‌تمرین) برگزار خواهد شد که تاریخ برگزاری آن‌ها از قبل اطلاع‌رسانی می‌شود. این کوییزها نقش مهمی در ارزیابی پیوسته پیشرفت شما دارند و به شما کمک می‌کنند نقاط ضعف خود را شناسایی و برطرف کنید.

- **مدیریت زمان:** پیشنهاد می‌شود زمان خود را به‌گونه‌ای مدیریت کنید که علاوه بر حضور فعال در کلاس، فرصت کافی برای مطالعه، انجام تمرین‌ها و مرور مطالب داشته باشید. یادگیری این درس نیازمند پیگیری منظم و مستمر است. همچنین حضور در کلاس‌های حل‌تمرین اجباری می‌باشد.
- **سیلابس درس:** سیلابس کامل درس در گیت‌هاب مربوط به این درس قرار داده شده است. برای اطلاع از بارم‌بندی بخش‌های مختلف و آشنایی با برنامه آموزشی ترم، حتماً به آن مراجعه کنید.
- **ارتباط با تیم آموزشی:** در صورت بروز هرگونه مشکل یا نیاز به مشورت، با سرپرست دستیاران آموزشی در ارتباط باشید. همچنین برای ارتباط مستقیم با استاد درس، می‌توانید از طریق ایمیل mo_vahidi@sbu.ac.ir اقدام کنید.

مطالعه بیشتر: تاریخچه و کاربردهای جاوا

زبان **جاوا** در اوایل دهه ۹۰ میلادی در زمانی متولد شد که صنعت نرم‌افزار با یک چالش جدی رو به رو بود: هر سیستم‌عامل و سخت‌افزار زبان و محیط توسعه تصمیم گرفت زبانی طراحی **Sun Microsystems** مخصوص به خود را داشت و انتقال یک نرم‌افزار از یک پلتفرم به دیگری هزینه و زمان زیادی می‌برد. شرکت و شعار مشهور «**یک‌بار بنویس، همه‌جا** (Java Virtual Machine - JVM)» کند که مستقل از سخت‌افزار و سیستم‌عامل باشد. نتیجه این تلاش ماشین‌محاذی جاوا **اجرا کن** بود. ایده اصلی این بود که برنامه‌ها به جای اجرا شدن مستقیم روی سیستم‌عامل، روی یک ماشین مجازی اجرا شوند که خودش با هر سیستم‌عامل هماهنگ می‌شود. این نوآوری باعث شد جاوا به سرعت جایگاه ویژه‌ای در توسعه نرم‌افزار پیدا کند، به خصوص در پروژه‌هایی که نیاز به پایداری، امنیت و قابلیت حمل بالا داشتند.

پیدایش (۱۹۹۰-۱۹۹۱)

را آغاز کرد. هدف اولیه، ساخت زبانی سیک و امن **Oak** پروژه‌ای با نام رمز به همراه تیمی از مهندسان **James Gosling**، در سال ۱۹۹۱ برای دستگاه‌های تعاملی و سیستم‌های توکار (مانند سستاپ‌باکس‌ها) بود. این زبان باید بتواند بدون وابستگی به سخت‌افزار، در محیط‌های مختلف اجرا شود. با

رشد اینترنت در اوایل دهه ۹۰، تیم توسعه متوجه شد که همین قابلیت «چندسکوئی» می‌تواند برای وب نیز بسیار ارزشمند باشد. به همین دلیل، در سال ۱۹۹۵ به جهان عرضه گردید (Applet) و تکنولوژی اپلت HotJava معرفی شد و همراه با مرورگر **Java** این زبان با نام جدید

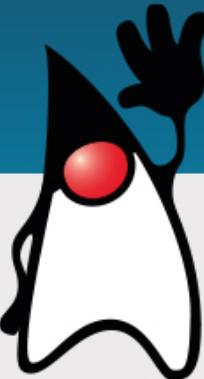
رشد چشمگیر کجا رخ داد؟

- جاوا به هسته اصلی سیستم‌های سازمانی در بانکداری، مخابرات و تجارت، **J2EE** و سپس **Servlet/JSP** **وب سازمانی (اواخر دهه ۹۰ و ۲۰۰۰)**: با معرفی الکترونیک تبدیل شد.
- باعث شد میلیون‌ها توسعه‌دهنده وارد اکوسیستم جاوا شوند و برنامه‌های موبایل خود را با این زبان بنویسند **اندروید (از ۲۰۰۸)**: عرضه.
- راه را برای توسعه سیستم‌های (Hadoop و Kafka مثل) **JVM** و فناوری‌های مبتنی بر **Spring Boot** **زیرساخت‌های مقیاس‌پذیر**: فریمورک‌هایی مانند ابری و مقیاس‌پذیر هموار کردند.

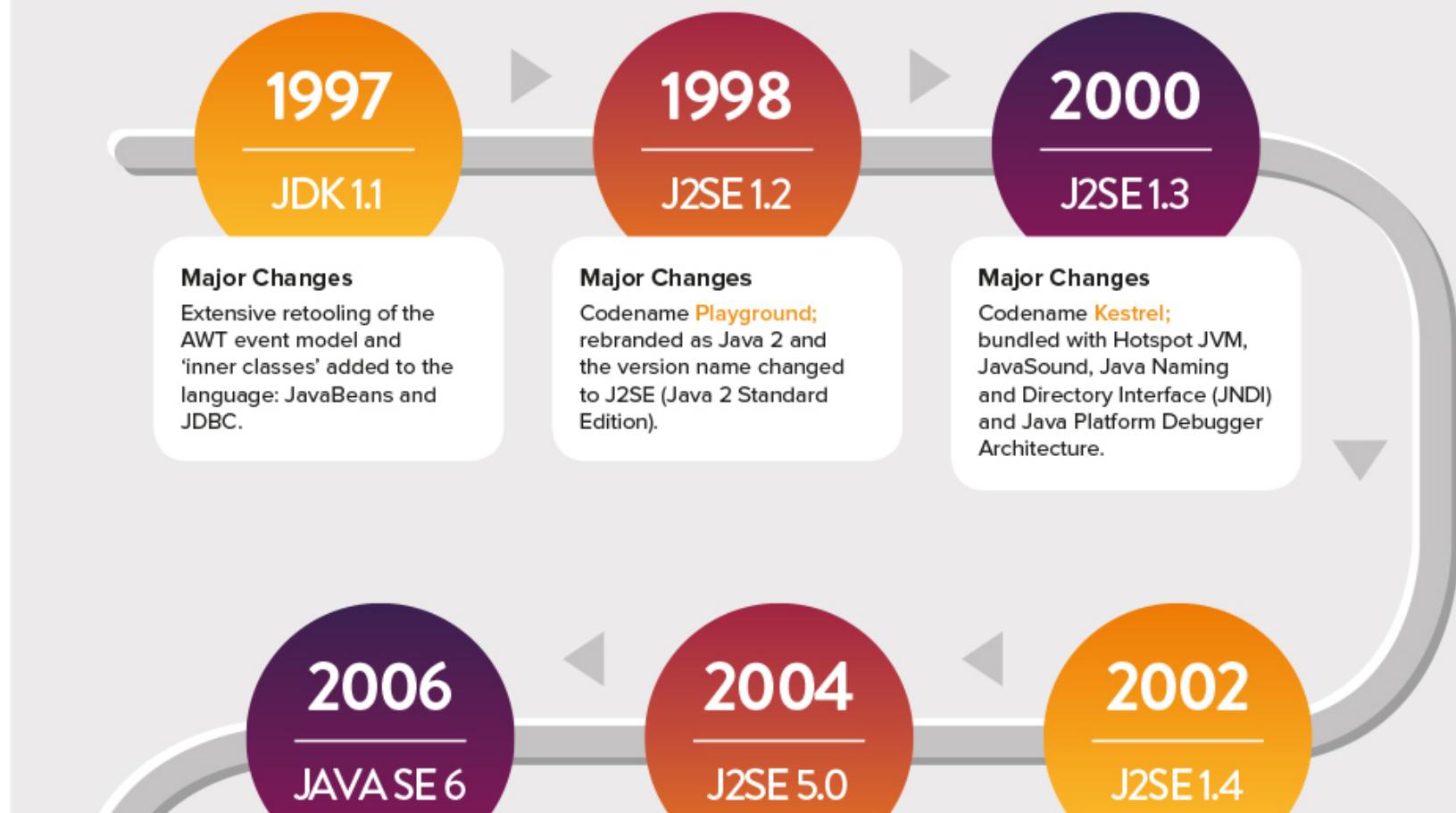
استفاده شرکت‌های بزرگ

- **Google**: بخش عمده اکوسیستم اندروید و سرویس‌های سمت سرور.
 - **Amazon**: سرویس‌های مقیاس‌پذیر و زیرساخت AWS.
 - **Netflix**: سیستم‌های پخش محتوا و معماری میکروسرویس.
-

A SHORT HISTORY OF JAVA



JAVA VERSION HISTORY







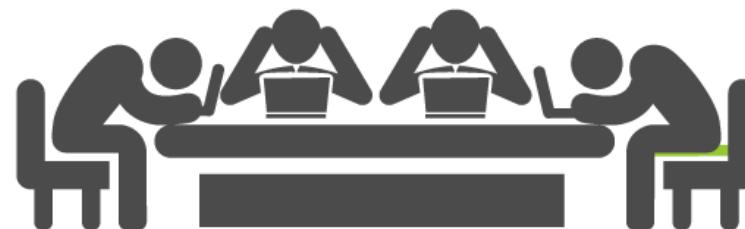
Java - history of Java programming language

Java celebrates 20 years on March 23rd 1995



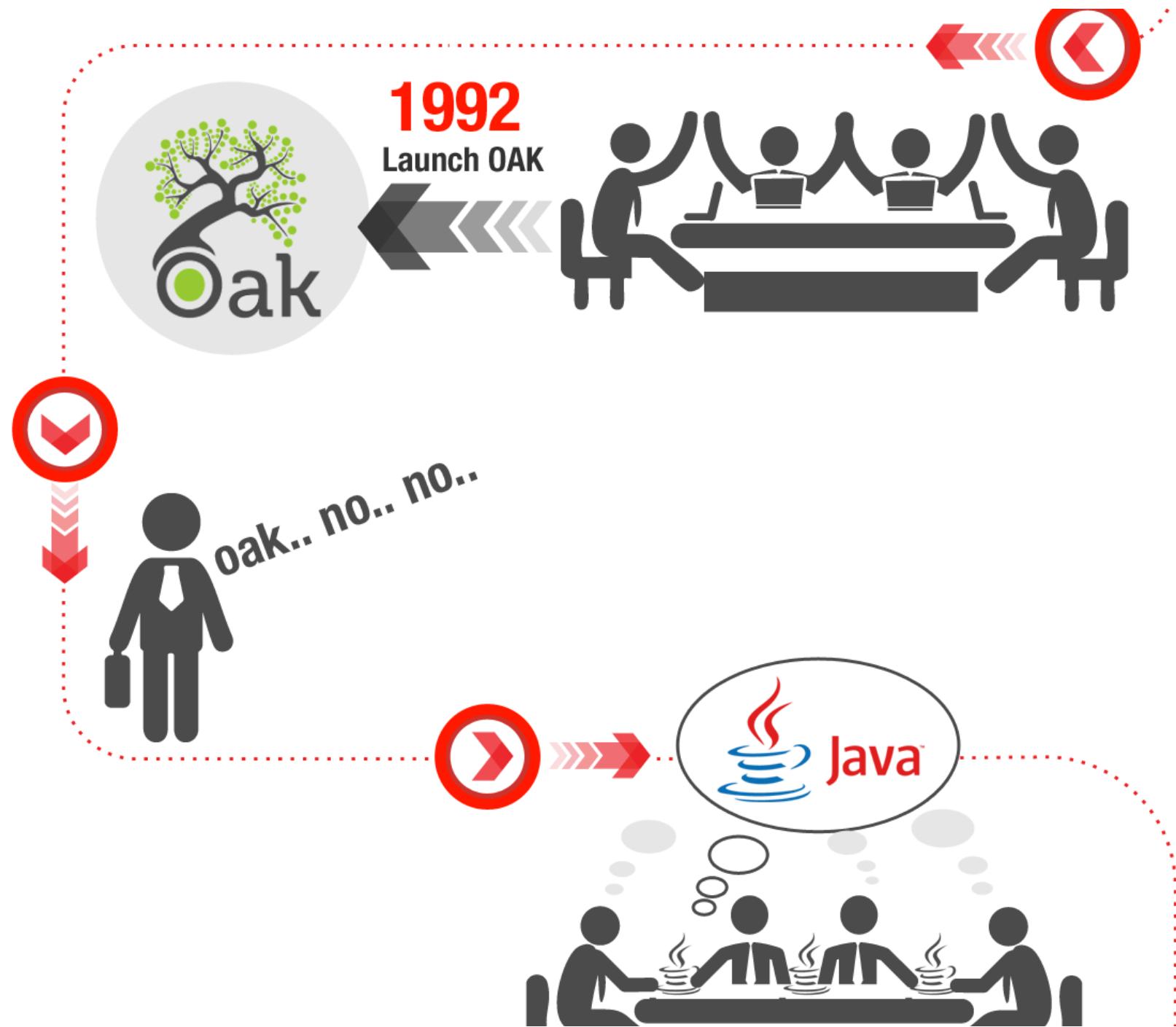
1991 - Green team with James Gosling tries to code for digital devices in C and C++.

They are slow...



Let's create a new language for that!





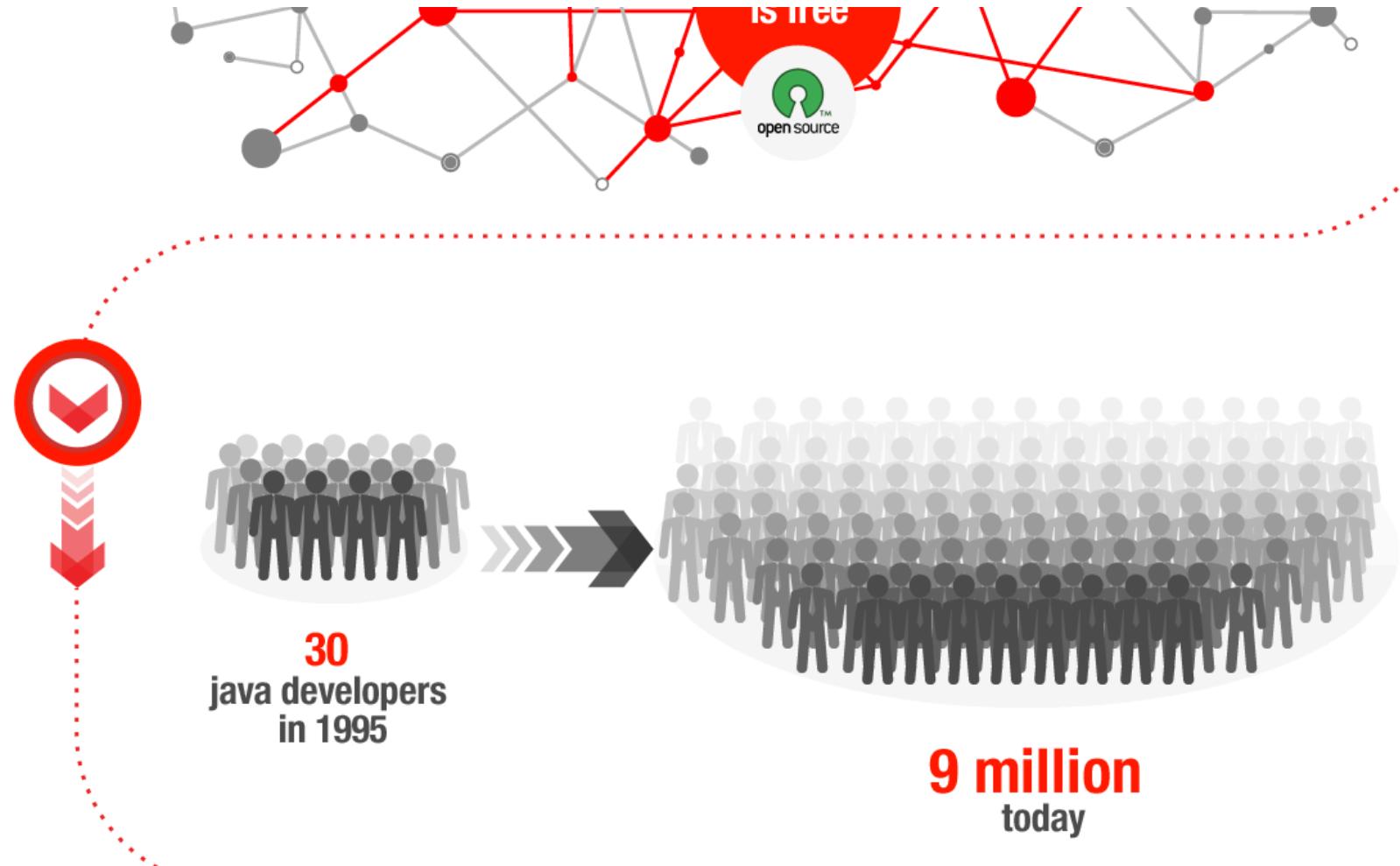
خودمون رو بسنجیم

این بخش برای این طراحی شده که در پایان مطالعه این اسلاید، بتونی خودت رو محک بزنی و بینی آیا مفاهیم رو به خوبی یاد گرفتی یا نه. سوالات زیر رو مرور کن و سعی کن بدون نگاه کردن به متن درس، به اون ها پاسخ بدی.



Java
is officially born





ORACLE®



Java
in 2010

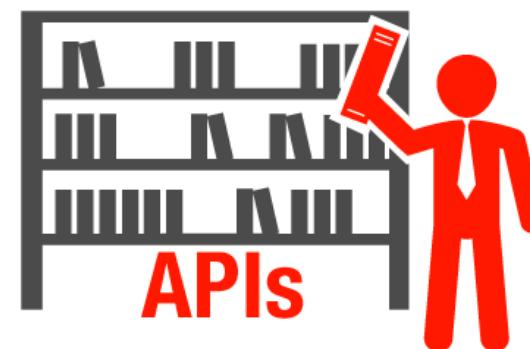




Java?

What's so great about it anyway?

Extensive libraries



APIs

Garbage collector

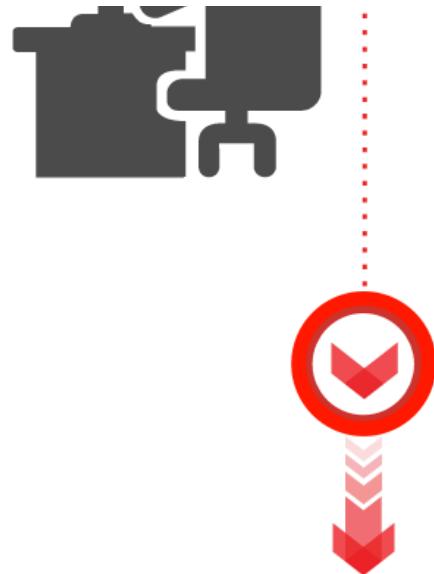


Multi-platform



Environment virtualization





Number 1
programming language
by developers community



9 million
Java developers



2 billion

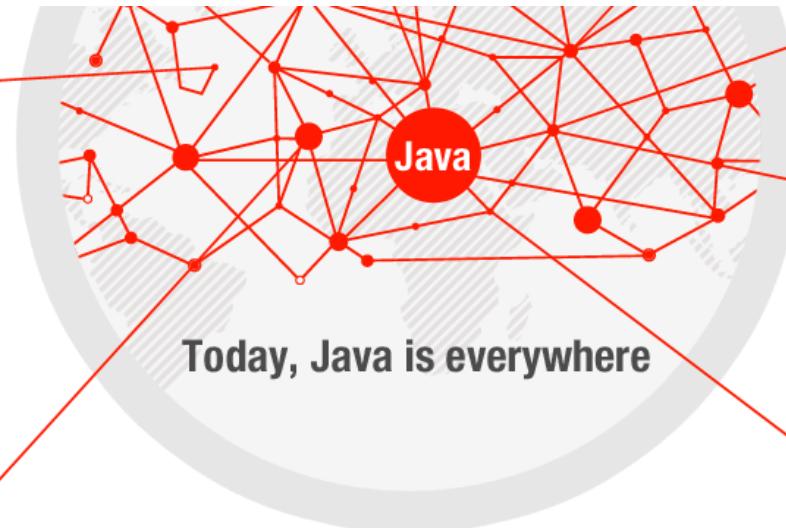


5 billion
Java cards in use



eXo
THE PLATEFORM

5 BILLION
mobile phones run Java



The Enterprise Social Platform

Number 1
business applications language

97%
of Enterprise Desktops Run Java

Language of choice
in all industries:



JAVA
Countries Distribution

