

عناصر برنامه نویسی: اولین برنامه‌ی جاوا

سید ناصر رضوی www.snrazavi.ir

۱۳۹۵



n.razavi@tabrizu.ac.ir

□ سید ناصر رضوی.

□ عضو گروه مهندسی کامپیوتر

□ سرپرست آزمایشگاه تحقیقاتی

«مدل سازی، شبیه سازی و یادگیری چند عاملی»

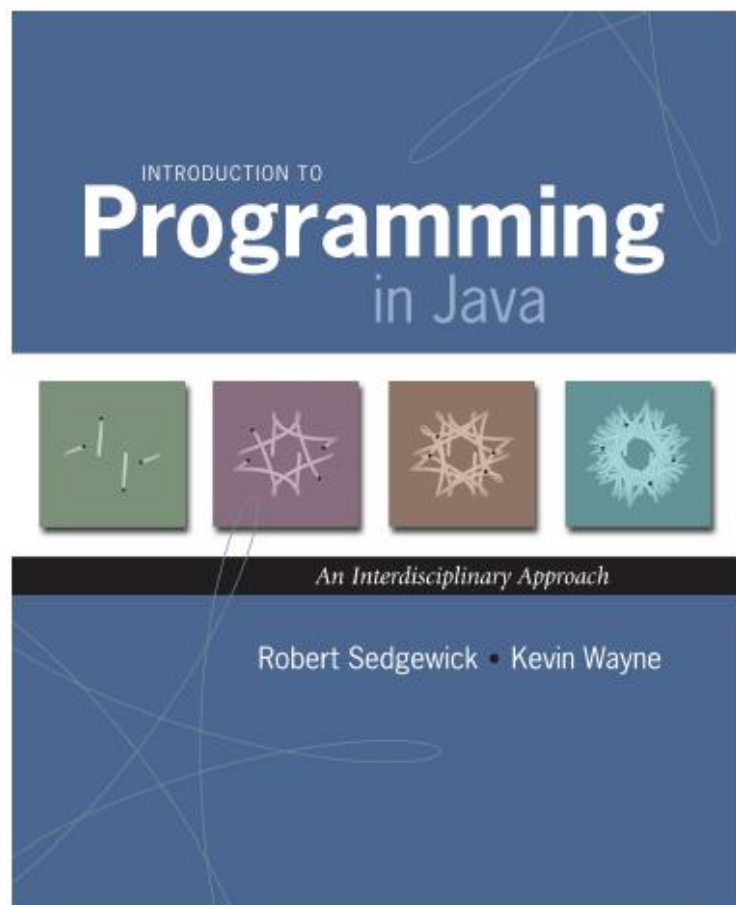
□ زمینه های پژوهشی.

□ هوش مصنوعی و یادگیری ماشین، یادگیری عمیق، یادگیری تقویتی

□ پردازش زبان طبیعی و ترجمه ماشینی

□ پردازش تصویر و بینایی ماشین

□ رباتیک



- مقدمه‌ای بر برنامه‌نویسی جاوا: یک رویکرد بین رشته‌ای
- رابرت سژویک و کوین وین، ۲۰۰۸

۱

عناصر برنامه‌نویسی.

- انواع داده‌ای پیش‌ساخته
- دستورات شرطی و تکرار
- آرایه‌ها
- ورودی و خروجی
- مطالعه‌ی موردی: کاوش تصادفی وب

۲

توابع و ماجول‌ها.

- متدهای استاتیک
- کتابخانه‌ها و برنامه‌های مشتری
- توابع بازگشتی
- مطالعه‌ی موردی: مسئله‌ی تراوش

۳

برنامه‌نویسی شی‌گرا.

- انواع داده‌ای
- ایجاد انواع داده‌ای جدید
- طراحی انواع داده‌ای
- مطالعه‌ی موردی: شبیه‌سازی چند جسمی

۴

الگوریتم‌ها و ساختمان داده‌ها.

- کارایی
- مرتب‌سازی و جستجو
- صف و پشته
- جداول نشانه
- مطالعه‌ی موردی: پدیده‌ی دنیای کوچک

ارزیابی

۵



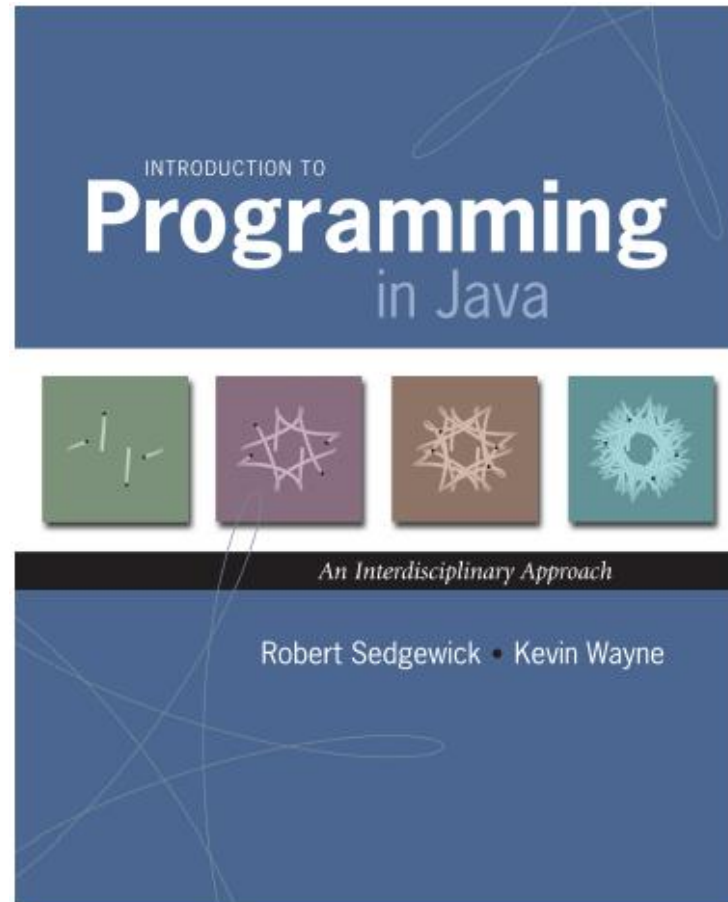
□ تمرین‌ها: ۲۰٪

□ پروژه‌ی کلاسی: ۲۰٪

□ آزمون پایان‌ترم: ۶۰٪

۱-۱ اولین برنامه‌ی شما

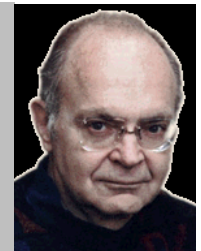
۶



زبان‌های برنامه‌نویسی

- زبان ماشین. خسته کننده و در معرض خطا.
- زبان‌های طبیعی. مبهم و درک و آنها برای کامپیوتر دشوار.
- زبان‌های برنامه‌نویسی سطح بالا. یک توازن قابل قبول!

« به جای آن که تصور کنید وظیفه اصلی شما این است که به کامپیوتر دستور دهید باید چه کار کند، بهتر است تمرکزمان بر روی این موضوع باشد که به انسان توضیح دهیم از کامپیوتر می‌خواهیم چه کاری انجام دهد.»
- دونالد کنوث



چرا برنامه‌نویسی؟

۹

□ چرا برنامه‌نویسی؟

- یک تجربه طبیعی، لذت بخش و خلاقانه.
- دستاوردهایی که بدون برنامه‌نویسی غیرممکن بودند.
- گشودن یک دنیای جدید از فعالیت‌های فکری و ذهنی.

□ اولین چالش. آموختن یک زبان برنامه‌نویسی.

□ پرسش بعدی. کدام زبان برنامه‌نویسی؟



انتخاب ما: جاوا

۱۰



جیمز گاسلینگ

<http://java.net/jag>

- ویژگی‌های جاوا.
 - استفاده‌ی گسترده
 - قابلیت دسترسی گسترده
 - شامل یک مجموعه‌ی کامل از انتزاع‌های مدرن
 - تنوع بررسی‌های خودکار برای یافتن خطاها در برنامه

□ صنعت جاوا.

۱۰۰ میلیارد دلار
۵ میلیون برنامه‌نویس

- مریخ نورد
- تلفن‌های موبایل
- دیسک‌های بلو-ری
- سرویس دهنده‌های وب
- تجهیزات پزشکی
- ...

چرا جاوا؟

□ ویژگی‌های جاوا.

- استفاده‌ی گسترده.
- قابلیت دسترسی گسترده.
- شامل یک مجموعه‌ی کامل از انتزاع‌های مدرن.
- تنوع بررسی‌های خودکار برای یافتن خطاها در برنامه.

□ حقیقت.

- هیچ زبان کاملی وجود ندارد.
- به هر حال باید **یک زبان** را انتخاب کنیم.

□ رویکرد ما.

- یک زیرمجموعه‌ی کمینه از جاوا.
- توسعه‌ی مهارت‌های عمومی برنامه‌نویسی که در بسیاری از زبان‌ها قابل استفاده هستند.

« تنها دو گونه زبان برنامه‌نویسی وجود دارد؛ یک دسته آنهایی که مردم همواره از آنها گله می‌کنند و دسته دیگر زبان‌هایی که هیچکس از آنها استفاده نمی‌کند. »

- بی‌یارنه استراس‌تروپ



یک زیرمجموعه غنی از جاوا

۱۲

Built-in Types	
<code>int</code>	<code>double</code>
<code>long</code>	<code>String</code>
<code>char</code>	<code>boolean</code>

Flow Control	
<code>if</code>	<code>else</code>
<code>for</code>	<code>while</code>

Boolean	
<code>true</code>	<code>false</code>
<code> </code>	<code>&&</code>
<code>!</code>	

String	
<code>+</code>	<code>""</code>
<code>length()</code>	<code>compareTo()</code>
<code>charAt()</code>	<code>matches()</code>

System
<code>System.out.print()</code>
<code>System.out.printf()</code>
<code>System.out.println()</code>

Parsing
<code>Integer.parseInt()</code>
<code>Double.parseDouble()</code>

Punctuation	
<code>{</code>	<code>}</code>
<code>(</code>	<code>)</code>
<code>,</code>	<code>;</code>

Arrays
<code>a[i]</code>
<code>new</code>
<code>a.length</code>

Math Library	
<code>Math.sin()</code>	<code>Math.cos()</code>
<code>Math.log()</code>	<code>Math.exp()</code>
<code>Math.sqrt()</code>	<code>Math.pow()</code>
<code>Math.min()</code>	<code>Math.max()</code>
<code>Math.abs()</code>	<code>Math.PI</code>

Primitive Numeric Types		
<code>+</code>	<code>-</code>	<code>*</code>
<code>/</code>	<code>%</code>	<code>++</code>
<code>--</code>	<code><</code>	<code>></code>
<code><=</code>	<code>>=</code>	<code>==</code>

Assignment
<code>=</code>

Objects	
<code>class</code>	<code>static</code>
<code>public</code>	<code>private</code>
<code>final</code>	<code>toString()</code>
<code>new</code>	<code>main()</code>

اولین برنامه: سلام، دنیا

۱۳



برنامه‌نویسی در جاوا: ایجاد برنامه

۱۴

□ برنامه‌نویسی در جاوا.

□ برنامه را با تایپ کردن آن در یک ویرایشگر متنی ایجاد کنید.

□ سپس آن را با نام HelloWorld.java ذخیره کنید.

```
/*  
 * Prints "Hello, World"  
 * Everyone's first Java program.  
 */  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

HelloWorld.java

برنامه‌نویسی در جاوا: کامپایل برنامه

□ برنامه‌نویسی در جاوا.

- برنامه را با تایپ کردن آن در یک ویرایشگر متنی ایجاد کنید.
- سپس آن را با نام HelloWorld.java ذخیره کنید.
- برنامه را با استفاده از دستور زیر در خط-فرمان **کامپایل** کنید.

فقط فرمان

```
% javac HelloWorld.java
```

- اجرای این فرمان باعث ایجاد یک فایل بایت کد جاوا به نام HelloWorld.class می‌شود.

برنامه‌نویسی در جاوا: اجرای برنامه

□ برنامه‌نویسی در جاوا.

- برنامه را با تایپ کردن آن در یک ویرایشگر متنی ایجاد کنید.
- سپس آن را با نام HelloWorld.java ذخیره کنید.
- برنامه را با استفاده از دستور زیر در خط-فرمان کامپایل کنید.
- برنامه را با اجرای دستور زیر در خط فرمان اجرا کنید.

خط فرمان

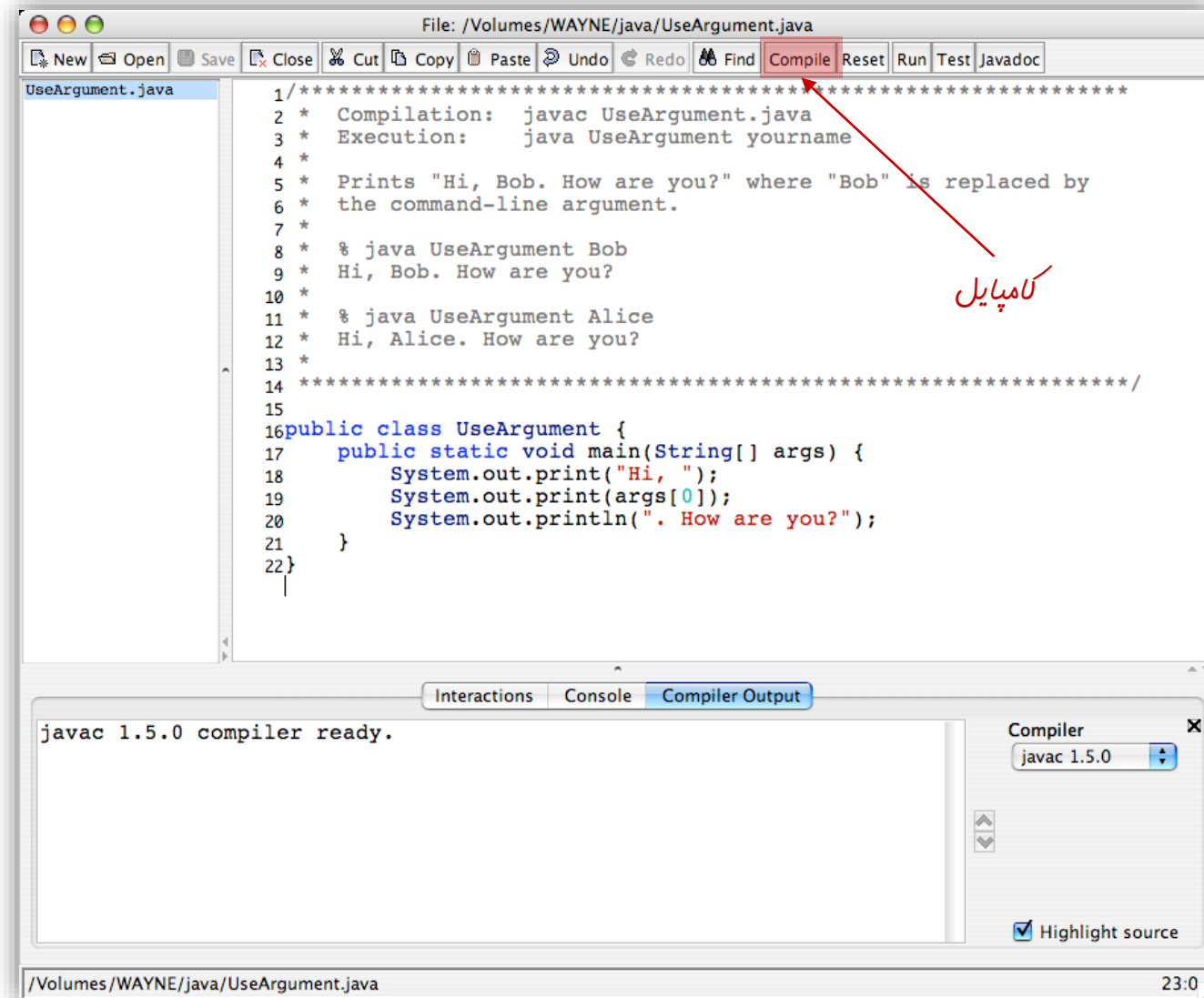
```
% javac HelloWorld.java  
  
% java HelloWorld  
Hello, World
```




<http://drjava.org>

دکتر جاوا

۱۸



```
File: /Volumes/WAYNE/java/UseArgument.java
New Open Save Close Cut Copy Paste Undo Redo Find Compile Reset Run Test Javadoc

UseArgument.java
1/*****
2 * Compilation: javac UseArgument.java
3 * Execution: java UseArgument yourname
4 *
5 * Prints "Hi, Bob. How are you?" where "Bob" is replaced by
6 * the command-line argument.
7 *
8 * % java UseArgument Bob
9 * Hi, Bob. How are you?
10 *
11 * % java UseArgument Alice
12 * Hi, Alice. How are you?
13 *
14 *****/
15
16public class UseArgument {
17    public static void main(String[] args) {
18        System.out.print("Hi, ");
19        System.out.print(args[0]);
20        System.out.println(". How are you?");
21    }
22}
```

کامپایل

Interactions Console **Compiler Output**

```
javac 1.5.0 compiler ready.
```

Compiler
javac 1.5.0

Highlight source

/Volumes/WAYNE/java/UseArgument.java 23:0

The screenshot shows the DrJava IDE interface. The top window displays the source code for `UseArgument.java`. The code includes a multi-line comment with instructions for compilation and execution, and a `main` method that prints a greeting with the first command-line argument.

```
1/*****
2 * Compilation: javac UseArgument.java
3 * Execution:   java UseArgument yourname
4 *
5 * Prints "Hi, Bob. How are you?" where "Bob" is replaced by
6 * the command-line argument.
7 *
8 * % java UseArgument Bob
9 * Hi, Bob. How are you?
10 *
11 * % java UseArgument Alice
12 * Hi, Alice. How are you?
13 *
14 *****/
15
16public class UseArgument {
17    public static void main(String[] args) {
18        System.out.print("Hi, ");
19        System.out.print(args[0]);
20        System.out.println(". How are you?");
21    }
22}
```

The bottom window shows the console output. The first two lines of the command prompt are highlighted with red boxes, and red arrows point from a handwritten Persian note to these lines.

```
Welcome to DrJava. Working directory is /Volumes/WAYNE/java
> java UseArgument Kevin
Hi, Kevin. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
> |
```

آرگومان‌های خط فرمان

File: /Volumes/WAYNE/java/UseArgument.java
New Open Save Close Cut Copy Paste Undo Redo Find Compile Reset Run Test Javadoc
Interactions Console Compiler Output
/Volumes/WAYNE/java/UseArgument.java 23.0