

برنامه نویسی سیستم

بخش دوم

```

class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick(View view) {
        Intent i = new Intent("net.learn");
        webView.setWebViewClient(new CallMyReceiver());
    }
}

```

دکتر اسماعیل صادق

esmaeilsadeghi.com

فهرت

۲	زبان برنامه نویسی جاوا.....
۳	کامپایلر های آنلاین جاوا.....
۴	تاریخچه.....
۴	مشخصات جاوا.....
۵	متغیرهای کلاس، شی، متدها و نمونه.....
۶	اولین برنامه جاوا.....
۷	کارکتر های خاص.....
۷	ایجاد، کامپایل و اجرای برنامه جاوا.....
۸	کلاس System.....
۸	متد getProperty.....
۹	کلاس Scanner.....
۱۰	کلاس File.....
۱۱	سوالات.....

زبان برنامه نویسی جاوا

جاوا یک زبان برنامه نویسی محبوب است که در سال ۱۹۹۵ ایجاد شد. این زبان متعلق به اوراکل است و بیش از ۳ میلیارد دستگاه جاوا را اجرا می کنند و استفاده می شود برای:

- برنامه های موبایل
- برنامه های دسکتاپ
- برنامه های کاربردی وب
- وب سرورها و سرورهای کاربردی
- بازی ها
- اتصال به پایگاه داده

جاوا بر روی پلتفرم های مختلف (ویندوز، مک، لینوکس، رزبری پای و غیره) کار می کند. این یکی از محبوب ترین زبان های برنامه نویسی در جهان است. در بازار کار فعلی تقاضای زیادی دارد. یادگیری آن آسان و استفاده از آن ساده است منبع باز و رایگان است. ایمن، سریع و قدرتمند است. این یک پشتیبانی اجتماعی عظیم دارد (ده ها میلیون توسعه دهنده). جاوا یک زبان شی گرا است که ساختاری واضح به برنامه ها می دهد و امکان استفاده مجدد از کد را فراهم می کند و هزینه های توسعه را کاهش می دهد. همچنین برای کار با جاوا نیازی به داشتن تجربه قبلی در برنامه نویسی نیست. برای بررسی اینکه آیا جاوا را روی رایانه شخصی ویندوزی نصب کرده اید، در نوار شروع جاوا را جستجو کنید یا عبارت زیر را در CMD تایپ کنید:

```
~ — -bash
~ $ java -version
java version "11.0.3" 2019-04-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.3+12-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.3+12-LTS, mixed mode)
~ $
```

جاوا تقاضای بسیار بالایی دارد و همه شرکت های بزرگ برنامه نویسان جاوا را برای توسعه برنامه های دسکتاپ، وب و موبایل خود استخدام می کنند. امروزه یک برنامه نویس جاوا با ۳ الی ۵ سال تجربه حدود ۱۲۰۰۰۰ دلار درخواست می کند و این زبان برنامه نویسی پرتقاضا در آمریکا است. شرکت های بزرگی که از جاوا استفاده می کنند و به برنامه نویسان خوب جاوا نیاز دارند در زیر آمده است:

گوگل
مایکروسافت
فیس بوک
IBM
آمازون
نتفلیکس
پینترست
جت برینز
و ...

یادگیری جاوا نسبتاً آسان است، بنابراین اگر شروع به یادگیری هر زبان برنامه نویسی کرده اید، جاوا می تواند انتخاب عالی شما باشد. همچنین ابزارهای جاوا زیادی وجود دارد که استفاده از آن را برای توسعه دهندگان و مبتدیان آسان می کند. دلایل خوب دیگری وجود دارد که جاوا را به عنوان اولین انتخاب هر برنامه نویسی تبدیل می کند:

- جاوا منبع باز است که به این معنی است که به صورت رایگان در دسترس است
- جاوا ساده است و یادگیری آن بسیار آسان است
- جاوا بسیار مورد تقاضا است و حقوق بالا را تضمین می کند
- جاوا ابزارهای توسعه قدرتمندی دارد
- جاوا مستقل از پلتفرم است
- جاوا یک زبان شی گرای می باشد

کامپایلر های آنلاین جاوا

برای کامپایل و اجرای کدهای خود در مرورگر، کامپایلر جاوا آنلاین را ارائه کرده اند. ما می توانید تقریباً تمام کدها را مستقیماً از مرورگر خود ویرایش و اجرا کنید. در زیر آدرس هایی برای اجرای کدهای جاوا از طریق مرورگر وجود دارد.

۱- <https://www.online-java.com/>

۲- <https://www.jdoodle.com/online-java-compiler/>

۳- https://www.w3schools.com/java/tryjava.asp?filename=demo_compiler

۴- https://www.tutorialspoint.com/online_java_compiler.php

۵- https://www.onlinegdb.com/online_java_compiler

تاریخچه

زبان برنامه نویسی جاوا در ابتدا توسط Sun Microsystems توسعه یافت که توسط جیمز گاسلینگ آغاز شد و در سال ۱۹۹۵ به عنوان جزء اصلی پلت فرم جاوا Java 1.0 J2SE منتشر شد.

مشخصات جاوا

۱. شی گرا: در جاوا همه چیز یک آبجکت^۱ است. جاوا را می توان به راحتی گسترش داد زیرا بر اساس مدل آبجکت است.
۲. مستقل از پلتفرم: برخلاف بسیاری از زبان های برنامه نویسی دیگر از جمله C و C++، زمانی که جاوا کامپایل می شود، در ماشین مخصوص پلتفرم کامپایل نمی شود، بلکه در کد بایت مستقل از پلتفرم کامپایل می شود. این کد بایت در وب توزیع شده و توسط ماشین مجازی^۲ بر روی هر پلتفرمی که روی آن اجرا می شود تفسیر می شود.
۳. ساده: جاوا برای یادگیری آسان طراحی شده است. اگر مفهوم اولیه OOP Java را درک کنید، تسلط بر آن آسان خواهد بود.
۴. امن: با ویژگی ایمن جاوا، امکان توسعه سیستم های عاری از ویروس و بدون دستکاری را فراهم می کند. همچنین تکنیک های احراز هویت مبتنی بر رمزگذاری کلید عمومی است.
۵. مستقل از معماری دستگاه^۳: بدین معنی است که نرم افزار توسعه یافته با Java طوری طراحی شده است که از محیط یا بستر اجرایی که قرار است در آن نصب و راه اندازی شود، کاملاً مستقل است. به طور مثال اندازه ی انواع داده ای اولیه این زبان در انواع محیط های سخت افزاری و نرم افزاری ثابت است. در زبان برنامه نویسی C، انواع داده ای int (عدد صحیح) در معماری ۳۲ بیتی ۲ بایت را به خود تخصیص می دهد و در معماری ۶۴ بیتی ۴ بایت را اشغال می کند. این در حالی است که نوع داده ای int زبان java در هر دو معماری مقداری یکسان از حافظه را اشغال می کند.
۶. قابل حمل: خنثی بودن از نظر معماری و نداشتن جنبه های وابسته به پیاده سازی مشخصات، جاوا را قابل حمل می کند.
۷. قوی: جاوا با تأکید بر بررسی خطای زمان کامپایل و زمان اجرا تلاش می کند تا موقعیت های مستعد خطا را از بین ببرد.
۸. چند رشته ای^۴: با ویژگی چند رشته ای جاوا می توان برنامه هایی نوشت که می توانند بسیاری از وظایف را به طور همزمان انجام دهند. این ویژگی طراحی به توسعه دهندگان اجازه می دهد تا برنامه های تعاملی را بسازند که می توانند به راحتی اجرا شوند.

¹ object

² jvm

³ architecture neutral

⁴ multithreaded

۹. تفسیر شده^۵: کد بایت جاوا به سرعت به دستورالعمل های ماشین بومی ترجمه می شود و در هیچ کجا ذخیره نمی شود.

۱۰. عملکرد بالا: با استفاده از کامپایلرهای Just-In-Time، جاوا کارایی بالایی را امکان پذیر می کند.

۱۱. توزیع شده: جاوا برای محیط توزیع شده اینترنت طراحی شده است.

۱۲. پویا: جاوا پویاتر از C یا ++C در نظر گرفته می شود زیرا برای انطباق با یک محیط در حال تکامل طراحی شده است. برنامه های جاوا می توانند حجم گسترده ای از اطلاعات زمان اجرا را حمل کنند که می توان از آنها برای تأیید و رفع دسترسی به اشیاء در زمان اجرا استفاده کرد.

متغیرهای کلاس، شی، متدها و نمونه

وقتی یک برنامه جاوا را در نظر می گیریم، می توان آن را مجموعه ای از اشیاء تعریف کرد که از طریق فراخوانی متدهای یکدیگر ارتباط برقرار می کنند. در ادامه به طور خلاصه ببینیم که متغیرهای کلاس، شی، متدها و نمونه به چه معنا هستند.

– شی^۶: اشیاء حالت ها و رفتارهایی دارند. مثال: سگ حالت هایی دارد، رنگ، نام، نژاد و همچنین رفتارهایی مانند تکان دادن دم، پارس کردن، غذا خوردن. یک شی نمونه ای از یک کلاس است.

– کلاس^۷: یک کلاس را می توان به عنوان یک الگو/طراحی تعریف کرد که رفتار/وضعیتی را که شی نوع آن پشتیبانی می کند، توصیف می کند.

– روش ها^۸: روش اساساً یک رفتار است. یک کلاس می تواند متدهای زیادی داشته باشد. در متدها منطق ها نوشته می شوند، داده ها دستکاری می شوند و تمام اقدامات اجرا می شوند.

– متغیرهای نمونه: هر شی دارای مجموعه منحصر به فرد خود از متغیرهای نمونه است. وضعیت یک شی توسط مقادیری که به این متغیرهای نمونه اختصاص داده شده است، ایجاد می شود.

⁵ interpreted

⁶ object

⁷ class

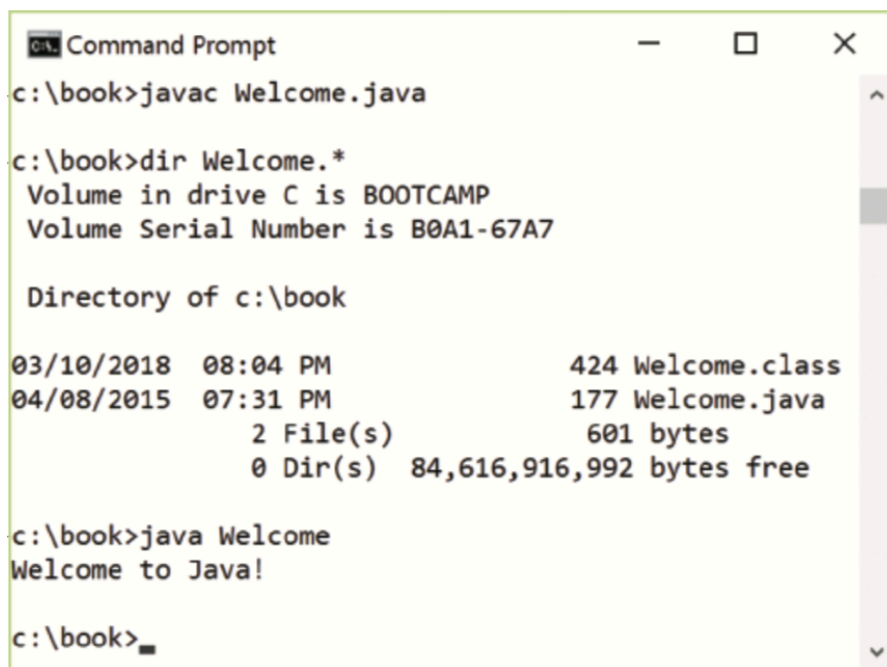
⁸ methods

اولین برنامه جاوا

یک برنامه جاوا از متد اصلی در کلاس اجرا می شود. بیایید نگاهی گذرا به مثال برنامه نویسی جاوا بیندازیم. در زیر مثال Welcome نوشته شده است.

```
public class Welcome {  
    public static void main(String args[]) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

و برای اجرا در cmd ویندوز به صورت زیر عمل می کنید:



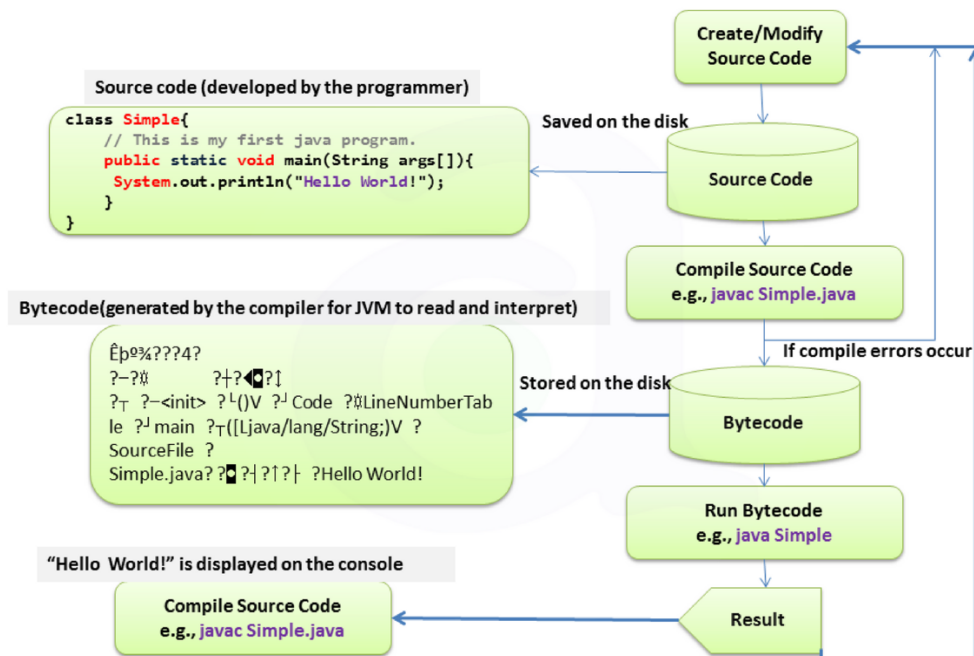
```
Command Prompt  
c:\book>javac Welcome.java  
  
c:\book>dir Welcome.*  
Volume in drive C is BOOTCAMP  
Volume Serial Number is B0A1-67A7  
  
Directory of c:\book  
  
03/10/2018  08:04 PM                424 Welcome.class  
04/08/2015  07:31 PM                177 Welcome.java  
                2 File(s)                601 bytes  
                0 Dir(s)  84,616,916,992 bytes free  
  
c:\book>java Welcome  
Welcome to Java!  
  
c:\book>
```

کارکترهای خاص

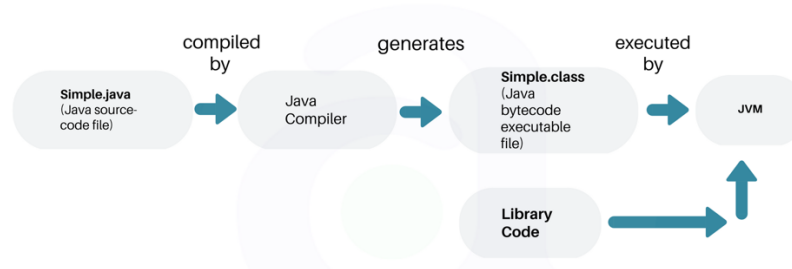
کارکتر	نام	توضیحات
{ }	باز و بسته شدن اکولاد	یک بلوک برای مشخص کردن عبارات استفاده می شود
()	باز و بسته شدن پرانتز	برای متدها استفاده می شود
[]	باز و بسته شدن براکت	یک آرایه را مشخص کنید
//	اسلش های دوتایی	برای نوشتن توضیحات
""	باز و بسته شدن گیومه	یک رشته (به عنوان مثال، دنباله ای از کاراکترها) را مشخص می کند
;	نقطه ویرگول	پایان یک خط کد را مشخصات می کند

ایجاد، کامپایل و اجرای برنامه جاوا

شما یک برنامه جاوا را در یک فایل با پسوند جاوا ذخیره می کنید و آن را در یک فایل با پسوند کلاس کامپایل می کنید. فایل کامپایل شده توسط ماشین مجازی جاوا اجرا می شود. پس بنابراین شما باید برنامه خود را بسازید و قبل از اجرا آن را کامپایل کنید. اگر برنامه شما دارای خطاهای کامپایل باشد، باید برنامه را تغییر دهید تا خطاها برطرف شوند، سپس آن را دوباره کامپایل کنید. اگر برنامه شما خطاهای زمان اجرا داشته باشد یا نتیجه درستی را ایجاد نکند، باید برنامه را دوباره تغییر دهید و آن را دوباره کامپایل کرده و دوباره اجرا کنید. شما می توانید از هر ویرایشگر متن یا IDE برای ایجاد و ویرایش یک فایل جاوا استفاده کنید.



کد های جاوا در بایت کد جاوا کامپایل می شود و بایت کد جاوا توسط ماشین مجازی جاوا تفسیر می شود. کد جاوا شما ممکن است از کد موجود در کتابخانه جاوا استفاده کند. ماشین مجازی جاوا کد شما را همراه با کد موجود در کتابخانه اجرا می کند.



برای اجرای یک برنامه جاوا، اجرای بایت کد برنامه است. شما می توانید بایت کد را روی هر پلتفرمی با ماشین مجازی جاوا که یک مفسر است اجرا کنید.

کلاس System

از جمله امکانات ارائه شده توسط کلاس System می توان به جریان های ورودی استاندارد، خروجی استاندارد و خروجی خطا اشاره کرد.

متد getProperty

برای دریافت ویژگی های سیستم که با کلید مشخص شده استفاده می شود و می توان آن ها را توسط این متد دریافت کرد

`static String getProperty(String key):`

نتیجه اجرای دستور بالا مقدار رشته ویژگی سیستم یا در صورتی که هیچ خاصیتی با آن کلید وجود نداشته باشد، null باشد. به عنوان مثال دستور زیر ویژگی های فعلی سیستم را تعیین می کند.

`System.getProperty()`

در جدول زیر بعضی کلید ها و مقادیر بازگشتی از سیستم مشخص شده است:

کلید	مقدار
os.name	اسم پلتفرم را بر می گرداند (به عنوان مثال: ویندوز یا مک)
os.arch	نوع سیستم عامل (به عنوان مثال: ۶۴ بیت یا ۳۲ بیتی)
os.version	نسخه سیستم عامل (به عنوان مثال: ۱۰.۱۳.۶)
user.dir	فهرست جاری کاربر
user.home	دایرکتوری اصلی کاربر
java.version	نسخه جاوا را بر می گرداند
java.vendor	فروشنده جاوا را بر می گرداند
java.vendor.url	آدرس فروشنده جاوا را بر می گرداند

ادامه این کلیدها در آدرس زیر قرار داده شده است:

– <https://docs.oracle.com/javase/8/docs/api/java/lang/System.html>

دستورات خروجی و ورودی می توان به صورت زیر باشد:

دستور	مقدار
System.out.println("Output Data");	نمایش خروجی داده در کنسول برنامه (یا CMD)
System.err.println("Output Error");	نمایش خروجی داده بصورت خطا در کنسول برنامه (یا CMD)
System.in.read();	دریافت ورودی داده در کنسول برنامه (یا CMD)

کلاس Scanner

در بعضی از موارد نیاز داریم تا بخشی از اطلاعات برنامه خود را از کاربر دریافت کنیم تا بر اساس اطلاعات ورودی، نتیجه حاصل از انجام پردازش های مد نظر روی داده های مربوطه در خروجی برگردانده شود که برای این منظور طراحان زبان برنامه نویسی جاوا کلاسی تحت عنوان Scanner طراحی کرده اند که چنین قابلیتی را فراهم می کند.

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

به عنوان مثال سیستم می توانیم یک خط را از کاربر دریافت کند و سپس آن را چاپ کند:

```
1 Scanner myScanner = new Scanner(System.in);
2 String userInput = myScanner.nextLine();
3 System.out.println(userInput);
4 myScanner.close();
```

در مثال فوق خط اول دستور تعریف کلاس Scanner را انجام می دهد، و در خط دوم از کاربر ورودی را دریافت می کند و همچنین در خط سوم ورودی که از کاربر دریافت کرده است را در خروجی چاپ می کند و در نهایت در خط چهارم جهت بسته شدن متغیر myScanner که جز منابع سیستمی می باشد، استفاده می شود.

کلاس File

کلاس File یک نمایش انتزاعی از نام مسیر فایل و دایرکتوری است. نام مسیر می تواند مطلق یا نسبی باشد. کلاس File چندین روش برای کار با دایرکتوری ها و فایل ها مانند ایجاد دایرکتوری ها یا فایل های جدید، حذف و تغییر نام دایرکتوری ها یا فایل ها، فهرست کردن محتویات دایرکتوری و غیره دارد.

متد	کاربرد
createNewFile()	ایجاد یک فایل در یک مسیر خاص
delete()	حذف فایل
isFile()	چک می کند که آیا مسیر داده شده، یک فایل است یا نه
isDirectory()	چک می کند که آیا مسیر داده شده، یک پوشه است یا نه
exists()	چک کردن اینکه آیا یک فایل در یک مسیر خاص وجود دارد یا نه
renameTo()	برای انتقال و تغییر نام یک فایل به کار می رود.
getName()	نام فایل را بر می گرداند.
getParent()	پوشه یا درایوی که فایل در آن قرار دارد را بر می گرداند.
getPath()	مسیر فایل را بر می گرداند.
length()	حجم فایل را بر حسب بایت بر می گرداند.
lastModified()	زمان آخرین دستکاری فایل را بر می گرداند.
mkdir()	برای ایجاد یک پوشه به کار می رود
mkdirs()	برای ایجاد یک پوشه های تو در تو به کار می رود

به عنوان مثال استفاده از متد createNewFile می توان یک فایل جدید را ایجاد کند. این متد یک مقدار بولی را بر می گرداند. اگر فایل ایجاد شود true و در غیر اینصورت false را بر می گرداند. البته قبل از استفاده از این متد باید یک نمونه از کلاس فایل ایجاد شده و سپس مسیری که قرار است فایل در آنجا ساخته شود به عنوان پارامتر به سازنده کلاس فایل ارسال شود. به عنوان مثال:

```
File f = new File("Test.txt");

if (f.createNewFile()) {
    System.out.println("The File is created!");
}
```

سوالات

۱. نحوه بررسی نصب جاوا بر روی سیستم.

جواب: در `cmd` یا `terminal` سیستم دستور `java -version` نوشته می شود و در صورت نصب بودن جاوا در سیستم مشخصات ورژن نصب شده نمایش داده می شود.

۲. نحوه نمایش زمان سیستم.

جواب:

```
System.out.println(new Date());
```

خروجی دستور فوق:

Mon Mar 13 20:50:47 IRST 2023

متدهای دستور `Date` :

کاربرد	متد
نمایش روز سیستم	<code>getDay()</code>
نمایش سال سیستم	<code>getYear()</code>
نمایش ماه سیستم	<code>getMonth()</code>
نمایش روز سیستم	<code>getDay()</code>
نمایش ساعت سیستم	<code>getHours()</code>
نمایش دقیقه سیستم	<code>getMinutes()</code>
نمایش ثانیه سیستم	<code>getSeconds()</code>

۳. مشخصات زبان برنامه نویسی جاوا را نام ببرید.

۴. متغیرهای کلاس، شی، متدها و نمونه را در زبان برنامه نویسی جاوا تعریف کنید.

۵. کاربرد کارکتر پرائتز در زبان برنامه نویسی جاوا چیست.
جواب: پرائتز برای متدها استفاده می شود.

۶. مراحل ایجاد، کامپایل و اجرای برنامه در زبان برنامه نویسی جاوا را توضیح دهید.

۷. کاربرد کلاس `System` چیست.

۸. چک کردن اینکه آیا یک فایل در یک مسیر خاص وجود دارد یا نه توسط کدام متد انجام می شود؟
جواب: `exists()`.

۹. چک کردن اینکه آیا مسیر داده شده، یک فایل است یا نه توسط کدام متد انجام می شود؟
جواب: `isFile()`.

۱۰. چک کردن اینکه آیا مسیر داده شده، یک پوشه است یا نه توسط کدام متد انجام می شود؟
جواب: `isDirectory()`.