

بنام خدا

برنامه نویسی جاوا

منابع:

- ۱- مقدمه کاملی بر برنامه نویسی شی گرا با جاوا: تألیف C.Thomas
- ۲- چگونه با java برنامه نویسی کنیم: تألیف Paul Dietel
- ۳- تفکر در جاوا: تألیف Prentice Hall
- ۴- آموزش جاوا در ۲۱ روز تألیف مهندس زارع پور

مقدمه

خصوصیات زبان Java

- (۱) جاوا در سال ۱۹۹۲ به بازار عرضه شده است.
- (۲) در جاوا برعکس C++، اشاره گر وجود ندارد و مدیریت حافظه برعهده خود سیستم است. به هر متغیر یا شی به محض ایجاد شدن، حافظه لازم داده شده و در پایان کار، حافظه داده شده آزاد می شود.

۳) شی گرا است: این زبان کلیه خصوصیات یک زبان شی گرا را دارا می باشد.
خصوصیات زبان شی گرا عبارتند از:
الف) کپسوله کردن (Encapsulation): بسته بندی خصوصیات و متدهای مربوط به یک موجودیت در یک بسته به نام Class.

ب) وراثت (inheritance): روشی برای استفاده از کد یک کلاس در کلاس های دیگر. مثلا مربع نوعی مستطیل است که طول و عرض آن برابر باشد، یا به عبارتی دیگر مربع از مستطیل ارث می برد یا مشتق می شود.

ج) چندریختی (Polymorphism): یعنی چند متد یا تابع می توانند نام یکسانی داشته باشند.

۴) Modular (ماژولار) است یعنی کد برنامه می تواند در بخش هایی با عنوان کلاس نوشته شود.

(۵) قابل حمل (Portable) است. یعنی اگر یک برنامه به زبان جاوا روی یک سیستم خاص نوشته باشید روی همه نوع سیستم اجرا می‌شود.

(۶) ایمن (Secure) است. هنگامی که از یک مرورگر وب استفاده می‌شود که شامل یک ریزبرنامه به زبان جاوا است. جاوا یک برنامه خاص جاوا را به محیط خاص اجرایی مربوطه‌اش منحصر کرده و اجازه دسترسی این برنامه را به سایر بخش‌های رایانه را نمی‌دهد یعنی با این کار، جلوی نفوذ برنامه‌ها و ویروس‌ها را می‌گیرد.

(۷) قدرتمند است و دستورات آن مشابه C و C++ است. جاوا سعی کرده است که از خطاهای موقع اجرا (Runtime Error) با اضافه کردن مدیریت آسان استنها (از جمله خطاهای "تقسیم بر صفر" و "File not found") جلوگیری کند.

(۸) چند نخ کشی شده (Multi threading) است. یعنی چندین زیر برنامه (نخ) می‌توانند همزمان در جاوا اجرا شوند.

۹) مستقل از پلت فرم (Platform) یا معماری خنثی است. یعنی یکبار که برنامه‌ای به زبان جاوا نوشته شد همیشه روی هر سیستمی اجرا بشود.

۱۰) جاوا برای اجرای برنامه‌ها با پسوند **Java**. ابتدا آنها را کامپایل کرده و یک فایل با پسوند **class**. که بصورت بایت کد می‌باشد ایجاد می‌کند. حال این فایل کامپایل شده برای اجرا باید تفسیر شده (با توجه به نوع سیستم عامل و cpu و ...) و اجرا شود. ضمناً پسوند فایل اجرایی بصورت **Jar**. خواهد بود.

نکته مهم: خروجی برنامه‌های C, C++ معمولاً یک فایل exe است درحالیکه خروجی برنامه‌های java کدهای قابل اجرا نیستند بلکه کدهای بایتی هستند که می‌توانند تفسیر شده و اجرا شوند.

انواع برنامه‌های جاوا

۱-Applet: از داخل صفحات وب اجرا می‌شود و دسترسی آن به منابع سیستم محدود است.

۲-Application: بصورت یک برنامه مستقل اجرا می‌شود و به تمام منابع سیستم دسترسی دارد.

نکته: برای اجرای برنامه‌های Application با زبان جاوا، باید یکی از نسخه‌های JRE یا همان **Java Runtime Environment** روی سیستم نصب شود (در اینجا نسخه ۷ یا همان 1.7). برای برنامه‌نویسی باید ابزار JDK یا همان Java Development Kit (در اینجا نسخه ۷) باید روی سیستم نصب شود.

نکته: برای نوشتن و اجرای برنامه به زبان جاوا دو روش وجود دارد:

روش ۱: ابتدا برنامه را در notepad نوشته و با پسوند java ذخیره کرده و سپس با استفاده از برنامه javac کامپایل کرده و برای اجرا باید از برنامه java کمک گرفت.

روش ۲: از یک IDE یا همان Integrated Development Environment مثل Eclipse استفاده کرده و برنامه را نوشته و سپس اجرا کرد.

مثال: ایجاد یک Applet

جواب: برای اینکار نرم‌افزار Eclipse را اجرا کرده و سپس از

File → New → Java Project

بعد روی src در Package Explorer کلیک راست و گزینه New و سپس class را

انتخاب می‌کنیم و اسم کلاس را Hello.java انتخاب می‌کنیم. (دستور import برای

اضافه کردن فایل‌های کتابخانه‌ای یا همان package ها به کلاس فعلی بکار می‌رود).

```
import java.applet.*;
import java.awt.Color;
import java.awt.Graphics;
public class Hello extends Applet{
    public void paint(Graphics g){
        g.drawString("Hello world", 10, 10);
    }
}
```

}

مقادیر ۱۰ ها مختصات نقطه شروع چاپ را نشان می دهند.
با دکمه F11 می توان Debug و یا با دکمه های Ctrl+F11 می توان پروژه را اجرا کرد.

بعد از اجرای این برنامه یک فایل Hello.class در مسیر workspace\Test\bin ساخته می شود. حال می خواهیم یک صفحه وب ایجاد کرده و این اپلت را تست کنیم.

```
<html>
<body>
<Applet>
  code="Hello.class"
</Applet>
</body>
</html>
```

مثال: ایجاد یک Application

مشابه قبل عمل کرده و بعد از وارد کردن کد زیر، آن را کامپایل و اجرا می کنیم.

```
public class Hello2 {
    public static void main(String[] args) {
        System.out.print("Hello World");
    }
}
```

نکته: اگر نخواهیم از IDE خاصی برای برنامه نویسی و اجرای برنامه های جاوا استفاده کنیم باید برنامه را در یک ویراستار متنی از جمله Notepad نوشته و بعد آن را با پسوند java. ذخیره می کنیم. سپس در Command Line با استفاده از برنامه

javac کامپایل کرده و با برنامه java آن را اجرا می‌کنیم (این برنامه ها در داخل JDK هستند).

تعریف کلاس:

نام [implements or [نام ابر کلاس extends] نام کلاس class شناسه کلاس
[واسط]
{
 ; تعریف متغیرها
 ; تعریف توابع (متدها)
}

شناسه کلاس (Class Modifier):

- ۱- public (عمومی): این کلاس توسط هر کلاس دیگر قابل دسترسی است.
- ۲- private (خصوصی): این کلاس توسط هر کلاس موجود در آن فایل قابل دسترسی است.
- ۳- abstract (انتزاعی): دارای متد انتزاعی می‌باشد. یک متد انتزاعی است اگر بدنه نداشته باشد. و بدنه آن در کلاس‌های به ارث برده شده تعریف می‌شود. البته کلاس انتزاعی می‌تواند شامل متد غیرانتزاعی با بدنه کامل باشد.
- ۴- final (نهایی): این کلاس نمی‌تواند کلاس فرزند داشته باشد.
- ۵- اگر خالی باشد برای کلاس‌های داخل همان package بصورت public در نظر می‌گیرد یعنی همه می‌توانند به آن کلاس دسترسی داشته باشند. ولی برای

package های دیگر بصورت private در نظر می گیرد یعنی آنها نمی توانند به این کلاس دسترسی داشته باشند.

کلمه extends برای ارث بری بکار می رود و مشخص می کند که کلاس فعلی از ابر کلاس ارث می برد.

کلمه implements اعلام کننده واسطی است که بایستی پیاده سازی شود. متدهای واسط در واقع بصورت کلی تعریف می شوند و جزئیات آن بعداً توسط بقیه کلاس ها پیاده سازی می شوند.

فرم کلی تعریف متغیر:

نام نوع شناسه

شناسه متغیر :

- ۱- public: توسط همه کلاس های دیگر قابل استفاده است.
- ۲- private: توسط خود کلاس قابل دسترسی است.
- ۳- protected: توسط خود کلاس و زیر کلاس ها قابل دسترسی است.
- ۴- static: بین تمام اشیاء ایجاد شده از کلاس، مشترک و یکسان خواهد بود و باید بوسیله نام کلاس به آن دسترسی داشت.
- ۵- final: مقدار آن نمی تواند توسط زیر کلاس های فرزند تغییر داده شود.

۶- اگر خالی باشد برای کلاس‌های داخل همان package بصورت public در نظر می‌گیرد یعنی همه می‌توانند به آن متغیر دسترسی داشته باشند. ولی برای package‌های دیگر بصورت private در نظر می‌گیرد یعنی آنها نمی‌توانند به این متغیر دسترسی داشته باشند.