

فهرست مطالب

بخش ۱: اصول اولیه جاوا

۲۴	فصل ۱: به جاوا خوش آمدید
۲۵	جاوا چیست و چرا این قدر عالی است؟
۲۵	مستقل از پلتفرم
۲۷	شی گرا بودن
۲۷	Java API
۲۸	اینترنت
۲۸	جاوا در برابر دیگر زبان های برنامه نویسی
۳۰	امکانات مهم زبان جاوا
۳۰	بررسی نوع
۳۱	مدیریت خودکار حافظه
۳۲	اداره استثناها
۳۲	نقاط ضعف جاوا
۳۳	تاریخچه کوتاهی از جاوا
۳۴	نام ها

۳۷	فصل ۲: نصب و استفاده از ابزارهای جاوا
۳۸	دانلود و نصب Java Development Kit
۳۸	دانلود JDK
۳۹	نصب JDK
۳۹	پوشه های JDK
۴۰	تعیین متغیر Path
۴۱	استفاده از ابزارهای خط فرمان جاوا
۴۱	کامپایل یک برنامه
۴۲	کامپایل بیش از یک فایل
۴۳	استفاده از گزینه های کامپایلر جاوا
۴۵	اجرای یک برنامه جاوا
۴۵	آشنایی با پیغام های خطا
۴۶	مشخص کردن گزینه ها
۴۶	

بخش ۲: اصول برنامه نویسی

۴۸	فصل ۱: اصول برنامه نویسی جاوا
۴۹	نگاهی به برنامه معروف Hello, World!
۵۲	کار کردن با کلمه های کلیدی

۵۴	کار کردن با دستورات
۵۴	انواع دستورات
۵۵	فاصله سفید
۵۶	کار کردن با بلاک‌ها
۵۷	ایجاد شناسه‌ها
۵۷	ایجاد توضیحات
۵۸	توضیحات تک خطی
۵۸	توضیحات چند خطی
۵۹	توضیحات JavaDoc
۵۹	آشنایی با برنامه‌نویسی شی‌گرا
۵۹	آشنایی با کلاس‌ها و اشیاء
۶۰	آشنایی با متدهای استاتیک
۶۰	ایجاد یک شی از یک کلاس
۶۱	دیدن یک برنامه که از یک شی استفاده می‌کند
۶۳	تفاوت چیست؟
۶۴	وارد کردن کلاس‌های API جاوا

فصل ۲: کار کردن با متغیرها و انواع داده

۶۸	معرفی متغیرها
۶۹	معرفی دو یا تعداد بیشتری متغیر در یک دستور
۶۹	معرفی متغیرهای کلاس
۷۰	معرفی متغیرهای نمونه
۷۰	معرفی متغیرهای محلی
۷۲	دادن مقدار به متغیرها
۷۲	مقدار دادن به متغیرها با دستورات نسبت‌دهی
۷۳	دادن مقدار به متغیرها با شروع کننده‌ها
۷۳	استفاده از متغیرهای نهایی (ثابت‌ها)
۷۴	کار کردن با انواع داده اصلی
۷۵	انواع عدد صحیح
۷۶	انواع دارای نقطه اعشار (یا اعداد اعشاری)
۷۸	نوع داده char
۷۹	نوع بولین (true/false)
۷۹	کلاس‌های پوششی
۸۰	استفاده از انواع ارجاعی
۸۱	کار کردن با رشته‌ها
۸۱	معرفی و مقداردهی رشته‌ها
۸۲	ترکیب کردن رشته‌ها
۸۲	تبدیل انواع اصلی به رشته‌ها
۸۳	تبدیل رشته‌ها به انواع اصلی

۸۳	تبدیل مقادیر عددی
۸۴	تبدیل خودکار
۸۴	تبدیل دستی نوع
۸۵	استفاده از باکسینگ
۸۵	آشنایی با قلمرو
۸۷	سایه‌ای کردن متغیرها
۸۹	چاپ داده‌ها با System.out
۸۹	استفاده از جریان‌های ورودی و خروجی استاندارد
۹۰	استفاده از System.out و System.err
۹۰	دریافت ورودی با کلاس Scanner
۹۱	وارد کردن کلاس Scanner
۹۲	معرفی و ایجاد یک شی Scanner
۹۲	دریافت ورودی
۹۴	دریافت ورودی کاربر با کلاس JOptionPane

فصل ۳: کار کردن با اعداد و عبارت‌ها ۹۷

۹۸	کار کردن با عملگرهای ریاضی
۱۰۰	تقسیم اعداد صحیح
۱۰۲	ترکیب عملگرها
۱۰۳	استفاده از عملگرهای مثبت و منفی یکانی
۱۰۴	استفاده از عملگرهای افزایشی و کاهشی
۱۰۶	استفاده از عملگر نسبت‌دهی (=)
۱۰۷	استفاده از عملگرهای نسبت‌دهی ترکیبی
۱۰۸	استفاده از کلاس Math
۱۰۸	استفاده از ثابت‌های کلاس Math
۱۰۹	کار کردن با توابع ریاضی
۱۱۲	ایجاد اعداد تصادفی
۱۱۵	توابع گرد کردن اعداد
۱۱۷	اعمال فرمت روی اعداد
۱۲۰	آشنایی با موارد غیرعادی درباره ریاضیات در جاوا
۱۲۰	سرریز شدن عدد صحیح
۱۲۱	مشکل اعداد اعشاری
۱۲۱	تقسیم بر صفر

فصل ۴: تصمیم‌گیری ۱۲۵

۱۲۶	استفاده از عبارت‌های بولین ساده
۱۲۸	استفاده از دستورات if
۱۲۸	دستورات if ساده
۱۳۰	دستورات if-else

۱۳۲		دستورات if تودرتو
۱۳۵		دستورات else-if
۱۳۷		استفاده از عملگرهای منطقی
۱۳۸		استفاده از عملگر !
۱۳۸		استفاده از عملگرهای & و &&
۱۳۹		استفاده از عملگرهای و
۱۴۰		استفاده از عملگر ^
۱۴۱		ترکیب عملگرهای منطقی
۱۴۳		استفاده از عملگر شرطی
۱۴۳		مقایسه رشته‌های متنی

فصل ۵: استفاده از حلقه‌ها برای تکرار دستورات ۱۴۵

۱۴۶		استفاده از حلقه while ساده
۱۴۶		دستور while
۱۴۶		یک حلقه شمارنده
۱۴۸		شکستن یک حلقه
۱۴۸		اجرای حلقه برای همیشه
۱۵۰		اجازه دادن به کاربر برای خاتمه دادن به حلقه
۱۵۰		اجازه دادن به کاربر از یک روش دیگر
۱۵۱		استفاده از دستور continue
۱۵۲		اجرای حلقه‌های do-while
۱۵۴		تایید اعتبار ورودی از کاربر
۱۵۷		استفاده از حلقه‌های for
۱۵۷		شکل رسمی حلقه for
۱۵۹		تعیین قلمرو متغیر شمارنده
۱۶۰		شمردن اعداد زوج
۱۶۱		شمارش برعکس
۱۶۲		استفاده از حلقه‌های for بدون بدنه آن
۱۶۳		ترکیب عبارت‌ها
۱۶۴		حذف عبارت‌ها
۱۶۴		شکستن و ادامه دادن در حلقه for
۱۶۵		تودرتو کردن حلقه‌ها
۱۶۵		یک حلقه تودرتوی ساده
۱۶۶		یک بازی حدس زدن

فصل ۶: استفاده از دستور switch ۱۷۱

۱۷۲		مشکلات دستورهای else-if
۱۷۲		یک مثال از یک برنامه else-if
۱۷۳		ایجاد یک نسخه بهتر از برنامه

۱۷۵	 استفاده از دستور switch
۱۷۶	 یک مثال switch
۱۷۷	 قرار دادن دستورات if درون دستورات switch
۱۷۸	 دستورات switch بر مبنای متغیر کاراکتری
۱۷۹	 استفاده نکردن از دستور break
۱۸۳	 استفاده از رشته‌های متنی درون switch

فصل ۷: اضافه کردن متد به برنامه‌ها ۱۸۵

۱۸۶	 لذت استفاده از متدها
۱۸۷	 اصول ایجاد متدها
۱۸۸	 یک مثال
۱۸۹	 یک مثال دیگر
۱۹۱	 متدهایی که مقادیری را بر می‌گردانند
۱۹۱	 معرفی نوع برگشتی متد
۱۹۲	 استفاده از دستور return برای برگرداندن مقدار برگشتی
۱۹۳	 استفاده از یک متد که یک نوع را بر می‌گرداند
۱۹۳	 باید یک دستور return مناسب داشته باشید
۱۹۵	 امتحان یک نسخه دیگر از برنامه بازی حدس زدن
۱۹۸	 متدهایی که پارامتر دریافت می‌کنند
۱۹۸	 معرفی پارامترها
۱۹۹	 قلمرو پارامترها
۲۰۰	 آشنایی با تحویل توسط مقدار
۲۰۰	 آزمایش یک نسخه دیگر از برنامه بازی حدس زدن

فصل ۸: اداره استثناها ۲۰۵

۲۰۶	 آشنایی با استثناها
۲۰۷	 دیدن یک استثنا
۲۰۷	 پیدا کردن متهم
۲۰۸	 گرفتن استثناها
۲۰۹	 یک مثال ساده
۲۱۰	 یک مثال دیگر
۲۱۲	 اداره استثناها به صورت مستقیم
۲۱۳	 دریافت یک‌جای همه استثناها
۲۱۴	 نمایش پیغام استثنا
۲۱۵	 استفاده از یک بلاک نهایی
۲۱۷	 اداره استثناهای بررسی شده
۲۱۸	 دیدن خطای کامپایلر دریافت-یا-انداختن
۲۱۹	 دریافت استثنای FileNotFoundException
۲۱۹	 انداختن یک استثنای FileNotFoundException
۲۲۰	 انداختن یک استثنا از متد main



۲۲۰	چشم‌پوشی از استثناها
۲۲۱	انداختن استثنای خود
۲۲۲	دریافت چند استثنا

بخش ۳: برنامه‌نویسی شی‌گرا

فصل ۱: آشنایی با برنامه‌نویسی شی‌گرا ۲۲۶

۲۲۷	برنامه‌نویسی شی‌گرا چیست
۲۲۷	آشنایی با اشیا
۲۲۷	اشیا دارای هویت هستند
۲۲۸	اشیا دارای نوع هستند
۲۲۸	اشیا دارای حالت نیز هستند
۲۲۹	اشیا دارای رفتار هستند
۲۳۰	آشنایی با چرخه عمر یک شی
۲۳۰	کار کردن با کلاس‌های مرتبط
۲۳۱	وراثت
۲۳۱	اینترفیس
۲۳۲	طراحی یک برنامه با اشیا
۲۳۳	ایجاد دیاگرام از کلاس‌ها با UML

فصل ۲: ایجاد کلاس ۲۳۵

۲۳۶	معرفی یک کلاس
۲۳۶	انتخاب نام کلاس
۲۳۶	بدنه کلاس
۲۳۷	دیدن یک کلاس
۲۳۹	کار کردن با اعضا
۲۳۹	آشنایی با فیلدها
۲۳۹	آشنایی با متدها
۲۴۰	سطح دسترسی
۲۴۰	استفاده از متدهای getter و setter
۲۴۲	بهبود عملکرد با متدهای هم نام
۲۴۳	ایجاد سازنده‌ها
۲۴۳	ایجاد سازنده‌های ساده
۲۴۴	ایجاد سازنده‌های پیش‌فرض
۲۴۵	فراخواندن سازنده‌های دیگر
۲۴۷	کاربردهای بیشتر برای کلمه کلیدی this
۲۴۸	استفاده از شروع کننده‌ها

فصل ۳: کار کردن با استاتیک‌ها ۲۴۹

۲۵۰	آشنایی با متدها و فیلدهای استاتیک
۲۵۱	کار کردن با فیلدهای استاتیک

۲۵۱	 استفاده از متدهای استاتیک
۲۵۲	 شمارش نمونه‌ها
۲۵۴	 استفاده از یک کلاس بدون ایجاد نمونه از آن
۲۵۵	 استفاده از شروع‌کننده‌های استاتیک

فصل ۴: استفاده از زیر کلاس‌ها و وراثت ۲۵۷

۲۵۸	 آشنایی با وراثت
۲۵۹	 موتورسیکلت و اتومبیل
۲۵۹	 یک مثال تجاری
۲۶۰	 سلسله مراتب وراثت
۲۶۰	 ایجاد زیر کلاس‌ها
۲۶۱	 بازنویسی متدها
۲۶۲	 حفاظت از اعضا
۲۶۳	 استفاده از this و super در زیر کلاس‌ها
۲۶۴	 درک وراثت و سازنده‌ها
۲۶۶	 استفاده از final
۲۶۶	 متدهای نهایی
۲۶۶	 کلاس‌های نهایی
۲۶۷	 تبدیل به بالا و پایین
۲۶۸	 تعیین نوع یک شی
۲۶۹	 پلی مورفیسم
۲۷۱	 ایجاد استثناهای دلخواه
۲۷۲	 زیر نظر داشتن سلسله مراتب انداختن
۲۷۳	 ایجاد یک کلاس استثنا
۲۷۴	 انداختن یک استثنای دلخواه

فصل ۵: استفاده از کلاس‌های انتزاعی و اینترفیس‌ها ۲۷۵

۲۷۶	 استفاده از کلاس‌های انتزاعی
۲۷۷	 استفاده از اینترفیس‌ها
۲۷۷	 ایجاد یک اینترفیس ساده
۲۷۸	 اجرای یک اینترفیس
۲۷۹	 استفاده از یک اینترفیس به عنوان یک نوع
۲۸۰	 کارهای دیگری که می‌توان با اینترفیس‌ها انجام داد
۲۸۰	 اضافه کردن فیلدها به یک اینترفیس
۲۸۱	 بسط دادن اینترفیس‌ها
۲۸۱	 استفاده از اینترفیس‌ها برای کالک

فصل ۶: استفاده از کلاس‌های Class و Object ۲۸۵

۲۸۶	 مادر تمام کلاس‌ها: Object
۲۸۶	 Object به شکل یک نوع
۲۸۷	 متدهای کلاس Object

۲۸۸	انواع اصلی شی نیستند
۲۸۹	متد toString
۲۸۹	استفاده از toString
۲۹۰	بازنویسی toString
۲۹۱	متد equals
۲۹۲	استفاده از equals
۲۹۳	بازنویسی متد equals
۲۹۶	متد clone
۲۹۷	پیاده‌سازی متد clone
۳۰۰	استفاده از clone برای ایجاد یک نسخه کامل
۳۰۱	ایجاد نسخه‌های عمیق
۳۰۶	کلاس Class

فصل ۷: ایجاد پکیج برای کلاس‌ها ۳۰۹

۳۱۰	کار کردن با پکیج‌ها
۳۱۰	وارد کردن کلاس‌ها و پکیج‌ها
۳۱۰	ایجاد پکیج‌ها
۳۱۲	یک مثال
۳۱۳	قرار دادن کلاس‌ها در یک فایل JAR
۳۱۴	گزینه‌های خط فرمان jar
۳۱۴	آرشیو کردن یک پکیج
۳۱۵	اضافه کردن یک فایل jar به ClassPath
۳۱۶	اجرای یک برنامه از یک آرشیو
۳۱۷	استفاده از Javadoc برای مستندسازی کلاس‌ها
۳۱۷	اضافه کردن توضیحات Javadoc
۳۲۰	استفاده از دستور javadoc
۳۲۰	دیدن صفحه‌های Javadoc

بخش ۴: رشته‌ها، آرایه‌ها و مجموعه‌ها

فصل ۱: کار کردن با رشته‌های متنی ۳۲۴

۳۲۵	نگاهی به رشته‌ها
۳۲۶	استفاده از کلاس String
۳۲۸	پیدا کردن طول یک رشته
۳۲۸	ایجاد تغییرات ساده در رشته
۳۲۹	استخراج کاراکترها از یک رشته
۳۳۰	استخراج زیررشته‌ها از یک رشته
۳۳۱	تقسیم کردن یک رشته
۳۳۳	عوض کردن قسمتی از یک رشته متنی

۳۳۴		StringBuffer و StringBuilder	استفاده از کلاس های
۳۳۵		StringBuilder	ایجاد یک شی
۳۳۵		StringBuilder	استفاده از متدهای
۳۳۷		StringBuilder	یک مثال از
۳۳۸		CharSequence	استفاده از اینترفیس

فصل ۲: استفاده از آرایه ها ۳۳۹

۳۴۰		آشنایی با آرایه ها
۳۴۱		ایجاد آرایه ها
۳۴۱		شروع یک آرایه (مقدار دادن اولیه)
۳۴۲		استفاده از حلقه های for برای آرایه ها
۳۴۳		حل تکالیف با آرایه ها
۳۴۵		استفاده از حلقه for بهبود یافته
۳۴۶		استفاده از آرایه ها با متدها
۳۴۷		استفاده از آرایه های دو بعدی
۳۴۷		ایجاد یک آرایه دو بعدی
۳۴۸		دسترسی به عناصر آرایه دو بعدی
۳۴۹		شروع یک آرایه دو بعدی
۳۵۰		استفاده از آرایه های ناهموار
۳۵۱		استفاده از ابعاد بیشتر
۳۵۲		یک مثال پیشرفته: یک صفحه شطرنج
۳۵۹		استفاده از کلاس Array
۳۶۱		پر کردن یک آرایه
۳۶۱		کپی کردن یک آرایه
۳۶۲		مرتب کردن یک آرایه
۳۶۲		جستجو درون یک آرایه
۳۶۳		مقایسه آرایه ها
۳۶۴		تبدیل آرایه ها به رشته های متنی

فصل ۳: استفاده از کلاس ArrayList ۳۶۵

۳۶۶		ArrayList	آشنایی با کلاس
۳۶۹		ArrayList	ایجاد یک شی
۳۷۰			اضافه کردن عناصر
۳۷۱			دسترسی به عناصر
۳۷۱			چاپ یک لیست آرایه ای
۳۷۲			استفاده از یک تکرارگر
۳۷۳			تغییر دادن عناصر
۳۷۵			حذف عناصر

فصل ۴: استفاده از کلاس LinkedList ۳۷۷

۳۷۸		LinkedList	آشنایی با کلاس
-----	-------	------------	----------------

۳۸۱	ایجاد یک لیست لینک شده
۳۸۱	اضافه کردن موارد به یک لیست لینک شده
۳۸۳	دریافت موارد از یک لیست لینک شده
۳۸۴	تغییر دادن موارد لیست لینک شده
۳۸۵	حذف موارد از لیست لینک شده

فصل ۵: ایجاد کلاس‌های مجموعه‌ای جنریک

۳۸۸	چرا از جنریک‌ها استفاده می‌شود؟
۳۸۹	ایجاد یک کلاس جنریک
۳۹۰	یک کلاس پشته‌ای جنریک
۳۹۳	استفاده از پارامترهای نوع وایلدکارد
۳۹۴	یک کلاس صف جنریک
۳۹۸	استفاده از عملگر لوزی

بخش ۵: تکنیک‌های برنامه‌نویسی

فصل ۱: برنامه‌نویسی ریسمان‌ها

۴۰۱	آشنایی با ریسمان‌ها
۴۰۲	ایجاد یک ریسمان
۴۰۲	آشنایی با کلاس Thread
۴۰۳	بسط کلاس Thread
۴۰۴	ایجاد و شروع یک ریسمان
۴۰۵	پیاده‌سازی اینترفیس Runnable
۴۰۵	استفاده از اینترفیس Runnable
۴۰۶	ایجاد کلاسی که Runnable را پیاده‌سازی کند
۴۰۷	استفاده از کلاس CountdownApp
۴۱۰	ایجاد ریسمان‌هایی که با یکدیگر کار می‌کنند
۴۱۳	استفاده از یک مجری
۴۱۶	متدهای همزمان شده
۴۲۰	ایجاد یک قفل
۴۲۱	کپی کردن با وقفه‌های ریسمان
۴۲۲	متوجه شدن این که متوقف شده‌اید
۴۲۳	خاتمه دادن به شمارش

فصل ۲: استفاده از عبارت‌های منظم

۴۳۰	ایجاد یک برنامه برای کار کردن با عبارت‌های منظم
۴۳۲	انجام انطباق کاراکتری ساده
۴۳۲	انطباق تک کاراکترها
۴۳۳	استفاده از کلاس‌های کاراکتری از پیش تعریف شده
۴۳۵	استفاده از کلاس‌های کاراکتری دلخواه
۴۳۶	استفاده از بازه‌ها
۴۳۷	منفی کردن

۴۳۷		منطبق کردن چند کاراکتر
۴۳۸		استفاده از کاراکترهای فرار
۴۳۹		استفاده از پرانتزها برای گروه‌بندی کاراکترها
۴۴۰		استفاده از نماد خط عمودی ()
۴۴۱		استفاده از عبارت‌های منظم در برنامه‌های جاوا
۴۴۱		مشکل String
۴۴۲		استفاده از عبارت‌های منظم در کلاس String
۴۴۲		استفاده از کلاس‌های Pattern و Matcher

فصل ۳: استفاده از بازگشتی‌ها ۴۴۵

۴۴۶		مثال محاسبه کلاسیک فاکتوریل
۴۴۶		راه حل بدون بازگشتی
۴۴۷		روش بازگشتی
۴۴۸		نمایش دایرکتوری‌ها
۴۵۱		نوشتن یک برنامه برای مرتب کردن
۴۵۲		درک چگونگی کارکرد مرتب‌سازی سریع
۴۵۲		استفاده از متد sort
۴۵۳		استفاده از متد partition
۴۵۵		قرار دادن همه موارد کنار یکدیگر

فصل ۴: استفاده از عبارت‌های لاندا و ارجاع‌های متد ۴۵۹

۴۶۰		آشنایی با عبارت‌های لاندا
۴۶۰		اصول عبارت‌های لاندا
۴۶۲		اینترفیس‌های تابعی
۴۶۳		استفاده از عبارت لاندا
۴۶۸		عبارت‌های لاندا ی بلاکی
۴۷۰		اینترفیس‌های تابعی جنریک
۴۷۲		دادن یک عبارت لاندا به عنوان یک آرگومان
۴۷۷		عبارت‌های لاندا و ثبت متغیر
۴۷۹		انداختن استثنا از درون یک عبارت لاندا
۴۸۰		ارجاع‌های متد
۴۸۰		ارجاع‌های متد به متدهای استاتیک
۴۸۲		ارجاع‌های متد به متدهای نمونه
۴۸۶		ارجاع‌های سازنده

بخش ۶: ایجاد رابط کاربری گرافیکی با سوییگ

فصل ۱: آشنایی با سوییگ ۴۹۰

۴۹۱		آشنایی با بعضی از مفاهیم اصلی سوییگ
۴۹۱		دیدن عملکرد سوییگ



۴۹۱		سلسله مراتب کلاس Swing
۴۹۳		فریم‌ها
۴۹۵		گفتن Hello, World! در سوییگ
۴۹۷		تعیین مکان فریم روی صفحه
۴۹۷		استفاده از کلاس JPanel
۴۹۹		استفاده از برجسب‌ها
۵۰۱		ایجاد دکمه‌ها
۵۰۲		کنترل چیدمان اجزا

فصل ۲: اداره رویدادها ۵۰۵

۵۰۶		آشنایی با رویدادها
۵۰۹		اداره کردن رویدادها
۵۱۱		ایجاد یک برنامه ClickMe
۵۱۳		استفاده از کلاس‌های درونی برای گوش دادن به رویدادها
۵۱۵		استفاده از عبارتهای لاندا برای اداره رویدادها
۵۱۸		اضافه کردن یک دکمه خروج
۵۱۹		دریافت رویداد WindowClosing
۵۲۲		نگاهی دوباره به برنامه ClickMe

فصل ۳: دریافت ورودی از کاربر ۵۲۵

۵۲۶		استفاده از فیلدهای متنی
۵۲۷		نگاهی به یک برنامه
۵۳۰		استفاده از فیلدهای متنی برای مقادیر عددی
۵۳۲		ایجاد یک کلاس تایید اعتبار
۵۳۳		استفاده از ناحیه‌های متنی
۵۳۴		کلاس JTextArea
۵۳۶		کلاس JScrollPane
۵۳۸		استفاده از چک‌باکس‌ها
۵۴۱		استفاده از دکمه‌های رادیویی
۵۴۳		استفاده از خط‌های دور
۵۴۵		استفاده از لغزنده‌ها
۵۴۸		طراحی یک برنامه سفارش پیتزا

فصل ۴: انتخاب از یک لیست ۵۵۳

۵۵۴		استفاده از کومبواکس‌ها
۵۵۵		ایجاد کومبواکس‌ها
۵۵۷		دریافت موارد از یک کومبواکس
۵۵۷		اداره کردن رویدادهای کومبواکس
۵۵۸		استفاده از لیست‌ها
۵۶۰		ایجاد یک لیست
۵۶۱		دریافت موارد از یک لیست

۵۶۳	تغییر موارد لیست
۵۶۴	استفاده از اسپنرها
۵۶۶	استفاده از درخت‌ها
۵۶۷	ایجاد یک درخت
۵۷۰	ایجاد یک جز JTree
۵۷۱	دریافت گره انتخاب شده
۵۷۲	نگاهی به یک برنامه کامل که از یک درخت استفاده می‌کند

فصل ۵: استفاده از مدیرهای طرح‌بندی ۵۷۷

۵۷۸	کار کردن با مدیرهای طرح‌بندی
۵۷۸	آشنایی با شش مدیر طرح‌بندی سوییچ
۵۷۹	اعمال مدیرهای طرح‌بندی
۵۷۹	استفاده از مدیر طرح‌بندی Flow
۵۸۰	استفاده از مدیر طرح‌بندی Border
۵۸۲	استفاده از مدیر طرح‌بندی Box
۵۸۵	استفاده از مدیر طرح‌بندی Grid
۵۸۶	استفاده از یک طرح‌بندی GridBag
۵۸۶	طراحی روی کاغذ
۵۸۷	اضافه کردن اجزا به یک طرح‌بندی GridBag
۵۸۹	کار کردن با سازنده‌های GridBagConstraints
۵۹۱	دیدن یک مثال طرح‌بندی GridBag
۵۹۴	استفاده از مدیر طرح‌بندی Group

فصل ۶: ریسمان‌های ورکر و کلاس SwingWorker ۵۹۹

۶۰۰	وظایف پس‌زمینه ساده
۶۰۹	یک مثال دیگر
۶۱۱	کنسل کردن وظیفه‌های پس‌زمینه

بخش ۷: استفاده از فایل‌ها و پایگاه داده

فصل ۱: کار کردن با فایل‌ها ۶۱۴

۶۱۵	استفاده از کلاس File
۶۱۵	آشنایی با متدها و سازنده‌های کلاس File
۶۱۷	ایجاد یک شی File
۶۱۸	ایجاد یک فایل
۶۱۸	دریافت اطلاعات درباره یک فایل
۶۱۹	دریافت محتویات یک دایرکتوری
۶۱۹	تغییر نام فایل‌ها
۶۲۰	حذف یک فایل
۶۲۱	استفاده از پارامترهای خط فرمان

انتخاب فایل‌ها در یک برنامه سوییچ	۶۲۲
ایجاد یک پنجره Open	۶۲۴
دریافت فایل انتخاب شده	۶۲۵
استفاده از فیلترهای فایل	۶۲۶
استفاده از اشیا Path	۶۲۸
ایجاد فایل‌ها با کلاس Paths	۶۲۹
دریافت محتویات یک دایرکتوری	۶۳۰
حرکت در یک درخت فایل در جاوای ۷ و بعد از آن	۶۳۱

فصل ۲: کار کردن با جریان‌های فایل

آشنایی با جریان‌ها	۶۳۶
خواندن جریان‌های کاراکتری	۶۳۷
ایجاد یک BufferedReader	۶۳۸
خواندن از یک جریان کاراکتری	۶۳۹
خواندن فایل movies.txt	۶۳۹
نوشتن جریان‌های کاراکتری	۶۴۳
وصل کردن یک PrintWriter به یک فایل متنی	۶۴۴
نوشتن در یک جریان کاراکتری	۶۴۶
نوشتن درون فایل	۶۴۶
خواندن جریان‌های باینری	۶۴۹
ایجاد یک DataInputStream	۶۵۱
خواندن از یک جریان ورودی داده	۶۵۱
خواندن فایل movies.dat	۶۵۳
نوشتن در جریان‌های باینری	۶۵۶
ایجاد یک DataOutputStream	۶۵۸
نوشتن درون یک جریان باینری	۶۵۸
نوشتن درون فایل movies.dat	۶۵۹

فصل ۳: استفاده از JDBC برای وصل شدن به یک پایگاه داده

نصب یک درایور	۶۶۴
نصب یک منبع داده ODBC	۶۶۴
نصب کانکتور MySQL در JDBC	۶۶۶
وصل شدن به یک پایگاه داده	۶۶۶
دادن کوئری به یک پایگاه داده	۶۶۸
اجرای یک دستور select	۶۶۹
حرکت درون سری نتایج	۶۷۰
دریافت داده‌ها از یک سری نتایج	۶۷۰
استفاده از تمام موارد: برنامه‌ای که از یک پایگاه داده استفاده می‌کند	۶۷۲
به روز کردن داده‌های SQL	۶۷۵

۶۷۶.....	استفاده از یک شی قابل به روز شدن RowSet
۶۷۸.....	حذف یک ردیف
۶۷۸.....	به روز کردن مقدار یک ستون برای ردیف مورد نظر
۶۷۹.....	قرار دادن یک ردیف

فهرست محتویات درون دیسک همراه کتاب

۹..... ادامه بخش ۷: کار کردن با فایل و پایگاه داده

۱۱..... فصل ۴: مبانی پایگاه داده

۱۲.....	تعریف یک پایگاه داده رابطه‌ای
۱۲.....	آشنایی با SQL
۱۳.....	آشنایی با دستورات SQL
۱۳.....	ایجاد یک پایگاه داده SQL
۱۶.....	دادن کوئری به یک پایگاه داده
۱۶.....	استفاده از دستور select
۱۸.....	محدود کردن کوئری
۱۸.....	مستثنی کردن ردیف‌ها
۱۹.....	استفاده از انتخاب‌های یگانه
۱۹.....	پیدا کردن چیزی که شبیه چیز دیگری است
۲۰.....	استفاده از توابع ستونی
۲۱.....	انتخاب داده‌ها از بیش از یک جدول
۲۲.....	از بین بردن تکرارها
۲۳.....	به روز کردن و حذف ردیف‌ها
۲۵.....	استفاده از دستور update

۲۷..... فصل ۵: استفاده از XML

۲۸.....	تعریف XML
۲۸.....	تگ‌ها
۲۹.....	خاصیت‌ها
۳۰.....	معرفی XML
۳۰.....	فایل movies.xml
۳۱.....	استفاده از یک DTD
۳۴.....	پردازش XML: استفاده از DOM و SAX
۳۵.....	خواندن یک سند DOM
۳۵.....	ایجاد یک DocumentBuilderFactory
۳۶.....	پیکربندی DocumentBuilderFactory
۳۶.....	ایجاد یک سازنده سند و خود سند
۳۷.....	استفاده از متد getDocument

۳۷	خواندن گره‌های DOM
۳۹	پردازش عناصر
۴۰	دریافت مقدار خاصیت‌ها
۴۱	دریافت مقدار عناصر فرزند
۴۱	استفاده از همه موارد: یک برنامه که فیلم‌ها را لیست می‌کند

بخش ۸: برنامه‌های جاوا برای وب و شبکه ۴۵

فصل ۱: ایجاد اپلت‌ها ۴۶

۴۷	آشنایی با اپلت‌ها
۴۸	کار کردن با کلاس JApplet
۴۹	نگاهی به یک اپلت نمونه
۵۲	ایجاد یک صفحه HTML برای یک اپلت
۵۳	آزمایش یک اپلت

فصل ۲: ایجاد سرولت‌ها ۵۵

۵۶	آشنایی با سرولت‌ها
۵۷	استفاده از تام‌کت
۵۷	نصب و پیکربندی تام‌کت
۵۹	اجرا و توقف تام‌کت
۶۰	آزمایش تام‌کت
۶۱	ایجاد یک سرولت ساده
۶۱	وارد کردن پکیج‌های سرولت
۶۱	بسط دادن کلاس HttpServlet
۶۲	چاپ روی صفحه وب
۶۲	پاسخ دادن با HTML
۶۵	اجرای یک سرولت
۶۵	بهبتر کردن سرولت HelloWorld
۶۷	دریافت ورودی از کاربر
۶۷	کار کردن با فرم‌ها
۶۸	استفاده از سرولت InputServlet
۷۰	استفاده از کلاس‌ها در یک سرولت

فصل ۳: استفاده از JSP ۷۵

۷۶	آشنایی با صفحات JavaServer
۷۷	استفاده از دایرکتیوهای صفحه
۷۸	استفاده از عبارات
۸۰	استفاده از اسکریپت‌ها
۸۲	استفاده از معرفی‌ها
۸۴	استفاده از کلاس‌ها

فصل ۴: برنامه‌نویسی شبکه ۸۹

۹۰	آشنایی با برنامه‌نویسی شبکه
۹۰	آدرس‌های IP و پورت‌ها
۹۱	نام‌هاست، DNS و URL‌ها
۹۲	تلنت
۹۲	دریافت اطلاعات درباره‌هاست‌های اینترنتی
۹۲	کلاس InetAddress
۹۴	یک برنامه برای جستجوی نام‌های هاست
۹۶	ایجاد برنامه‌های سرور شبکه
۹۶	کلاس Socket
۹۸	کلاس ServerSocket
۹۹	برنامه BART
۹۹	کلاس BartQuote
۱۰۱	برنامه BartServer
۱۰۴	برنامه BartClient
۱۰۷	کار کردن با BartSvrer 2.0
۱۱۱	نوشتن دستورات معمول جاوا
۱۱۲	نوشتن انواع داده اصلی
۱۱۲	کلاس‌های Math و NumberFormat
۱۱۴	استفاده از عملگرهای جاوا

PDF Compressor Free Version

بخش ۱

اصول اولیه جاوا

PDF Compressor Free Version

بخش ۱

فصل ۱

به جاوا خوش آمدید

راهنمای جامع
و کاربردی جاوا



این فصل یک آشنایی خیلی کوچک با دنیای جاوا است. در چند صفحه بعدی، متوجه می‌شوید که جاوا چیست، از کجا آمده و به کجا می‌رود. همچنین بعضی از مواردی که باعث قدرتمندتر شدن جاوا می‌شوند را به همراه بعضی از نقطه ضعف‌های آن خواهید دید. به علاوه می‌بینید که جاوا در مقایسه با دیگر زبان‌های برنامه‌نویسی پرتعداد مانند C، C++ و C# چگونه است.

در ضمن من در این فصل فرض می‌کنم که حداقل می‌دانید که برنامه‌نویسی کامپیوتر چیست. این بدان معنی نیست که شما را یک متخصص یا برنامه‌نویس حرفه‌ای در نظر می‌گیرم، بلکه بدان معنی است که زمان برای توضیح مفاهیم ساده‌ای مانند یک برنامه کامپیوتری چیست، یک زبان برنامه‌نویسی چیست و مانند آن نمی‌گذارم.

در این فصل با قطعه‌های کد جاوا برخورد خواهید کرد که به همراه تعدادی کد نوشته شده در زبان‌های دیگر مانند C، C++ و بیسیک به شما اجازه مقایسه عملکرد این زبان‌ها را می‌دهند. اگر مفهوم این کدها را متوجه نمی‌شوید نترسید. من فقط می‌خواهم حسی از ظاهر و چگونگی برنامه‌های جاوا به دست آورده و بتوانید آن را با دیگر زبان‌های برنامه‌نویسی مقایسه نمایید.

نکته

تمام کدهای لیست شده در این کتاب را می‌توانید در دیسک همراه کتاب پیدا کنید. این کدها برحسب فصل و بخش تقسیم شده‌اند.

به علاوه به علت کمبود فضا چند فصل تکمیلی دیگر نیز درون دیسک همراه کتاب قرار داده شده است.

جاوا چیست و چرا این قدر عالی است؟

جاوا یک زبان برنامه‌نویسی مشابه با C و C++ است. به عنوان یک نتیجه، اگر با C یا C++ کار کرده باشید، با اغلب امکانات جاوا آشنا خواهید بود. (برای اطلاعات بیشتر شباهت‌ها و تفاوت‌های بین جاوا و C و C++ به قسمت “جاوا در برابر دیگر زبان‌های برنامه‌نویسی” در همین فصل مراجعه کنید.)

اما جاوا دارای تفاوت‌های مهمی با دیگر زبان‌های برنامه‌نویسی است. من مهم‌ترین تفاوت‌ها را در قسمت‌های بعدی ذکر می‌کنم.

مستقل از پلتفرم

یکی از مهم‌ترین دلایل محبوبیت زیاد جاوا، مستقل از پلتفرم (Platform Independence) بودن آن است، یعنی برنامه‌های جاوا را می‌توان روی انواع مختلفی از کامپیوترها اجرا کرد. یک برنامه جاوا از طریق یک محیط/اجرای جاوا (Java Runtime Enviroment) روی هر کامپیوتری اجرا می‌شود که به صورت خلاصه با نام یک JRE شناخته می‌شود. یک JRE را می‌توانید برای انواع مختلف کامپیوترها دریافت کنید: کامپیوترهای تحت ویندوز، مکینتاش، یونیکس و لینوکس، کامپیوترهای مین‌فریم بزرگ و حتی روی تلفن‌های همراه.

قبل از جاوا، دیگر زبان‌های برنامه‌نویسی با فراهم آوردن کامپایلرهای سازگار برای پلتفرم‌های مختلف سعی در فراهم آوردن استقلال از پلتفرم داشتند. یک کامپایلر (Compiler) برنامه‌ای است که کدهای نوشته شده در یک زبان برنامه‌نویسی را به فرمی تبدیل می‌کند که بتواند واقعاً روی کامپیوتر اجرا شود. ایده کار این بود که بتوانید نسخه‌های مختلفی از برنامه‌ها را برای هر پلتفرم کامپایل (Compile) کنید (یعنی برنامه را به فرمی در بیاورید که قابل اجرا روی کامپیوتر باشد). متأسفانه، این ایده هیچ وقت عملی نشد. کامپایلرها هیچ وقت روی همه پلتفرم‌ها یکسان نبودند؛ هرکدام دارای تفاوت‌های جزئی با یکدیگر بودند. به عنوان یک نتیجه، باید نسخه‌های مختلفی از برنامه خود را برای هر پلتفرمی که می‌خواستید پشتیبانی شود، فراهم می‌آوردید.

اما مستقل از پلتفرم بودن جاوا بر مبنای کامپایلرهای سازگار با پلتفرم‌های مختلف نیست. جاوا روی مفهوم “ماشین مجازی” یا Virtual Machine متکی است. می‌توانید یک ماشین مجازی جاوا (که بعضی وقت‌های JVM نامیده می‌شود) را به عنوان یک پلتفرم کامپیوتری مجازی در نظر بگیرید- یعنی نوعی طراحی برای یک کامپیوتر که دارای هیچ سخت‌افزار واقعی نیست. در حقیقت JRE یک شبیه‌ساز است- یعنی برنامه‌ای است که روی هارد دیسک شما مشابه با یک کامپیوتر عمل می‌کند (یعنی JVM) که می‌تواند برنامه‌های جاوا را اجرا نماید.

کامپایلر جاوا کدهای جاوا را به زبان ماشین کامپیوتر تبدیل نمی‌کند تا اجرا شوند. بلکه، کدهای جاوا را به زبان ماشین JVM تبدیل می‌کند که بایت‌کد (Bytecode) نام دارند. سپس JRE بایت‌کد را در JVM اجرا می‌کند. به دلیل استفاده از JVM، می‌توانید یک برنامه جاوا را روی هر کامپیوتری که JRE روی آن نصب است، اجرا نمایید بدون این که به کامپایل مجدد برنامه خود نیاز داشته باشید.

به همین دلیل است که جاوا مستقل از پلتفرم می‌باشد- و باور کنید یا نه، به خوبی نیز این کار را انجام می‌دهد. برنامه‌هایی که می‌نویسید نه تنها روی همه کامپیوترهای PC با هر نسخه‌ای از ویندوز اجرا می‌شوند بلکه به راحتی روی مکینتاش، یونیکس یا لینوکس یا هر کامپیوتر دیگری که دارای JRE نصب شده باشد، اجرا می‌گردند.

چند مورد دیگر نیز برای این استقلال از پلتفرم وجود دارد:

◀ برنامه JRE از کامپایلر جاوا مستقل است. به عنوان یک نتیجه، مجبور نیستید تا یک کامپایلر جاوا را نصب کنید تا بتوانید برنامه‌های جاوای کامپایل شده را اجرا نمایید. تمام چیزی که نیاز دارید، JRE است.

◀ وقتی شخصی از شما درباره داشتن جاوا سؤال می‌کند، معمولاً منظور او این است: آیا محیط اجرای جاوا (JRE) نصب است؟ یعنی اگر JRE را داشته باشید می‌توانید برنامه‌های جاوا را اجرا کنید.

◀ مستقل بودن از پلتفرم برای همه کامپیوترها نیست. به عنوان مثال اگر یک سیستم کامپیوتر غیرعادی داشته باشید- مانند یک Olivetti Programma 101 عتیقه- و JRE نیز برای آن موجود نباشد، نمی‌توانید برنامه‌های جاوا را روی آن اجرا نمایید.

نکته تکنیکی

من عبارت Olivetti Programma 101 را از خودم نساختم. این یک کامپیوتر رومیزی بود که در اوایل دهه ۱۹۶۰ ساخته شد و مقدمه ورود من به برنامه‌نویسی کامپیوتری بود. (دبیر ریاضی دبیرستان من یکی از آن‌ها را در دبیرستان داشت و به من اجازه می‌داد در زمان ناهار از آن استفاده کنم.) با یک جستجو در گوگل می‌توانید چندین وب سایت درباره آن پیدا نمایید.

◀ مستقل بودن از پلتفرم جاوا کامل نیست. با این که بایت‌کد روی هر کامپیوتری که دارای JRE باشد اجرا می‌شود اما بعضی از قسمت‌های جاوا از سرویس‌های مخصوص سیستم عامل استفاده می‌کند. به عنوان یک نتیجه، بعضی وقت‌ها تفاوت‌های کوچکی به خصوص در برنامه‌هایی که از اینترفیس گرافیکی استفاده می‌کنند، ظاهر می‌شود.

◀ به دلیل این که یک سیستم اجرا که یک JVM را شبیه‌سازی می‌کند مأمور اجرای بایت‌کد جاوا می‌باشد، بعضی افراد به اشتباه جاوا را با زبان‌های برنامه‌نویسی ترجمه‌ای (Interperated) مانند بیسیک یا پرل (Perl) مقایسه می‌کنند. این زبان‌ها اصلاً کامپایل نمی‌شوند. بلکه مترجم آن‌ها (Interpreter) در زمان