

آموزش برنامه نویسی جاوا

جلد ۱

ترجمه و تألیف: انور پوراحمد

انتشارات پندار پارس

سرشناسه	: پوراحمد، انور، ۱۳۶۹ -
عنوان و نام پدیدآور	: آموزش برنامه‌نویسی جاوا/ انور پوراحمد.
مشخصات نشر	: تهران: پندار پارس: مانلی، ۱۳۹۰ -
مشخصات ظاهری	: ج: مصور، جدول.
شابک	: ۱۱۰۰۰۰ ریال با لوح فشرده: جلد اول: 978-964-2989-66-9 دوره: 3-964-2989-978
وضعیت فهرست نویسی	: فیپا
موضوع	: جاوا (زبان برنامه‌نویسی کامپیوتر)
رده بندی کنگره	: QA۷۶/۷۳ ج۲پ۹ ۱۳۹۰
رده بندی دیویی	: ۰۰۵/۱۳۳
شماره کتابشناسی ملی	: ۸۴۲۵۰۳۲

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگرجنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
 تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸
info@pendarepars.com



نام کتاب	: آموزش برنامه‌نویسی جاوا (جلد ۱)
ناشر	: انتشارات پندار پارس ناشر همکار: مانلی
ترجمه و تالیف	: انور پور احمد
چاپ اول	: بهار ۹۰
شمارگان	: ۱۰۰۰ نسخه
طرح جلد	: محمد اسماعیلی هدی
لیتوگرافی، چاپ، صحافی	: ترام‌سنج، صالحان، نوین برتر
قیمت	: ۱۱۰۰۰ تومان به همراه DVD شابک : ۹۷۸-۹۶۴-۲۹۸۹-۶۶-۹

شابک دوره : ۹۷۸-۹۶۴-۲۹۸۹-۶۸-۳



*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

فهرست

فصل ۱	برنامه نویسی چیست؟	۱
۲,۱	ساختمان کامپیوتر (آناتومی کامپیوتر)	۳
۷	وقایع تصادفی ۱,۱: ENIAC و آغاز محاسبات (پیدایش کامپیوترها)	۷
۱,۳	تبدیل برنامه‌های قابل فهم انسان به زبان ماشین	۸
۱,۴	زبان برنامه نویسی جاوا	۱۰
۱,۵	آشنایی با کامپیوتر	۱۲
۱,۵	اشاره سوئمنند ۱,۱: درک فایل‌های سیستم	۱۵
۱,۶	اشاره سوئمنند ۱,۲: داشتن یک خط مشی (استراتژی) برای پشتیبان‌گیری	۱۶
۱,۶	کامپایل یک برنامه ساده	۱۷
۱,۱	خطای رایج ۱,۱: از قلم انداختن نقطه ویرگول (;	۲۳
۱,۱	مبحث پیشرفته ۱,۱: دستورالعمل‌های معمول و متداول	۲۴
۱,۷	خطاها	۲۵
۱,۲	خطاهای رایج ۱,۲: غلط‌های املائی	۲۷
۱,۸	پروژه کامپایل	۲۷
برنامه نویسی	پروژه های برنامه نویسی	۳۵
فصل ۲	استفاده از آبجکت‌ها	۳۷
۲,۱	انواع متغیرها	۳۸
۲,۲	عملگر انتساب	۴۰
۲,۳	آبجکت‌ها، کلاس‌ها و متدها	۴۲
۲,۴	پارامترهای متدها و مقادیر بازگشتی	۴۵
۲,۵	انواع ارقام	۴۹

۵۱.....	۲,۶ ساخت آبجکت‌ها
۵۳.....	خطای رایج ۲,۱: ایجاد سازنده‌های مانند متد
۵۳.....	۲,۷ متدهای دسترسی و تحول
۵۴.....	۲,۸ آزمایش برنامه‌ها
۵۷.....	مبحث پیشرفته ۲,۱: آزمایش کلاس‌ها در محیط تعاملی
۵۸.....	۲,۹ اسناد API
۶۱.....	اشاره سودمند ۲,۱: استفاده از اینترنت
۶۱.....	۲,۱۰ مرجع آبجکت‌ها
۶۵.....	وقایع تصادفی ۲,۱: پردازنده‌های مرکزی
۶۶.....	۲,۱۱ برنامه‌های گرافیکی و چهارچوب پنجره‌ها
۶۹.....	۲,۱۲ ترسیم و اجزای آن
۷۳.....	مبحث پیشرفته ۲,۲: اپلت‌ها
۷۶.....	۲,۱۳ اشکال بیضی شکل، خطوط، متون و رنگ‌ها
۷۸.....	رنگ‌ها
۸۱.....	وقایع تصادفی ۲,۲: سیر تکاملی اینترنت
۹۳.....	پروژه‌های برنامه‌نویسی
۹۹.....	فصل ۳ اجرای کلاس‌ها
۹۹.....	۳,۱ سطوح انتزاعی
۱۰۳.....	۳,۲ مشخص کردن رابط عمومی کلاس
۱۰۹.....	۳,۳ تعبیر اینترفیس public
۱۱۲.....	اشاره سودمند ۳,۱: استفاده سودمند از javadoc
۱۱۲.....	۳,۴ فیلدهای نمونه
۱۱۵.....	۳,۵ اجرای سازنده‌ها و متدها

۱۱۹.....	۳,۱: نحوه ایجاد کلاس.....
۱۲۳.....	۳,۶ واحد آزمایش.....
۱۲۵.....	اشاره سودمند ۳,۲: استفاده از خط فرمان بصورت مؤثر.....
۱۲۶.....	۳,۷ انواع متغیرها.....
۱۲۸.....	۳,۸ پارامترهای متدی ضمنی و آشکار.....
۱۳۱.....	مبحث پیشرفته ۳,۱: فراخوانی سازنده.....
۱۳۱.....	وقایع تصادفی ۳,۱: دستگاه‌های رأی گیری الکترونیکی.....
۱۳۴.....	۳,۹ کلاس‌های اشکال.....
۱۳۹.....	۳,۲ نحوه ترسیم اشکال گرافیکی.....
۱۴۴.....	وقایع تصادفی ۳,۲: گرافیک کامپیوتر.....
۱۸۵.....	فصل ۴ انواع داده‌ای اصلی.....
۱۸۵.....	۴,۱ انواع داده‌ای.....
۱۸۹.....	مبحث پیشرفته ۴,۱: اعداد بزرگ.....
۱۹۰.....	مبحث پیشرفته ۴,۲: اعداد دودویی (Binary).....
۱۹۱.....	وقایع تصادفی ۴,۱: اشکال اعداد ممیز شناور در پنتیوم‌ها.....
۱۹۳.....	۴,۲ ثوابت.....
۱۹۸.....	نکته مدیریتی ۴,۱: از اعداد شگفت‌انگیز استفاده نکنید.....
۱۹۸.....	نکته مدیریتی ۴,۲: از نام‌های توصیفی برای متغیرها استفاده کنید.....
۱۹۹.....	۴,۳: تخصیص، افزایش و کاهش.....
۲۰۰.....	اشاره سودمند ۴,۱: جلوگیری از ترتیب(چیدمان) ناپایدار.....
۲۰۱.....	مبحث پیشرفته ۴,۲: ترکیب عملگر جایگزینی و محاسباتی.....
۲۰۲.....	عملگرهای محاسباتی و توابع ریاضی.....
۲۰۶.....	خطای رایج ۴,۱: تقسیم صحیح.....

خطای رایج ۴,۲: پیرانتزهای نامتوازن.....	۲۰۶
نکته مدیریتی ۴,۳: فاصله سفید (فضای خالی).....	۲۰۷
نکته مدیریتی ۴,۴: فاکتورگیری از کدهای معمول و رایج	۲۰۸
۴,۵: فراخوانی متدهای Static (ایستا)	۲۰۸
خطای رایج ۴,۳: خطاهای گردکردن (رند کردن)	۲۱۰
توضیح ۴,۱: انجام محاسبات	۲۱۱
۴,۶ رشته‌ها	۲۱۴
اشاره سودمند ۴,۲: مطالعه بر روی گزارشات استثناها (Exception).....	۲۱۷
مبحث پیشرفته ۴,۴: دستورات گریز (Escape)	۲۱۸
مبحث پیشرفته ۴,۵: رشته‌ها و انواع char(کاراکتر).....	۲۲۰
وقایع تصادفی ۴,۲: حروف و الفبای بین‌المللی	۲۲۱
مبحث پیشرفته ۴,۶: قالب بندی اعداد	۲۲۵
مبحث پیشرفته ۴,۷: استفاده از جعبه‌های محاوره‌ای برای ورودی و خروجی‌ها.....	۲۲۷
فصل ۵ تصمیمیات	۲۴۵
۵,۱ دستور if	۲۴۵
نکته مدیریتی ۵,۱: طرح‌بندی براکت‌ها	۲۴۹
اشاره سودمند ۵,۱: تورفتگی‌ها و تب‌ها	۲۵۰
مبحث پیشرفته ۵,۱: عملگر انتخابی	۲۵۱
۵,۲ مقایسه مقادیر	۲۵۲
خطای رایج ۵,۱: استفاده از == جهت مقایسه رشته‌ها	۲۵۷
نکته مدیریتی ۵,۲: اجتناب از اثرات جانبی	۲۵۸
۵,۳ گزینه‌های چندگانه	۲۵۸
اشاره سودمند ۵,۲: میانبرهای صفحه کلید	۲۶۴

۲۶۶.....	اشاره سوومند ۵,۳: کپی و جایگذاری در محیط ویرایشگر (IDE)
۲۶۶.....	مبحث پیشرفته ۵,۲: دستور switch
۲۶۸.....	خطای رایج ۵,۲: رهایی از مشکلات ناشی از دستور else
۲۶۹.....	اشاره سوومند ۵,۴: مدیریت زمانبندی مشکلات غیر منتظره
۲۷۰.....	مبحث پیشرفته ۵,۳: انواع شمارشی
۲۷۲.....	۵,۴ استفاده از عبارات بولین
۲۷۶.....	خطای رایج ۵,۳: عملگرهای رابطهای چندگانه
۲۷۷.....	خطای رایج ۵,۴: شرایط گیج کننده && و
۲۷۷.....	مبحث پیشرفته ۵,۴: ارزیابی عملگرهای بولین
۲۷۸.....	مبحث پیشرفته ۵,۵: قانون دمورگان
۲۷۹.....	۵,۵ انجام آزمایشات
۲۸۰.....	نکته مدیریتی ۵,۳: محاسبه داده‌ها بصورت دستی
۲۸۱.....	نکته مدیریتی ۵,۴: پیش‌بینی حالت‌های آزمایش از قبل
۲۸۲.....	مبحث پیشرفته ۵,۶: واقعه نگاری
۳۰۹.....	فصل ۶ تکرار ۳۰۹
۳۰۹.....	۶,۱ حلقه while
۳۱۴.....	خطای رایج ۶,۱: حلقه‌های نامتناهی
۳۱۵.....	خطای رایج ۶,۲: خروج با یک خطا
۳۱۶.....	مبحث پیشرفته ۶,۱: حلقه‌های do
۳۱۷.....	وقایع تصادفی ۶,۱: کد اسپاگتی
۳۲۰.....	۶,۲ حلقه for
۳۲۴.....	نکته مدیریتی ۶,۱: استفاده از حلقه‌ها با توجه به اهداف
۳۲۵.....	خطای رایج ۶,۳: از قلم انداختن نقطه ویرگول (;)

خطای رایج ۶,۴: استفاده نابجا از نقطه ویرگول(;).....	۳۲۶
نکته مدیریتی ۶,۲: برای آزمایش انتهای دامنه از != استفاده نکنید	۳۲۷
مبحث پیشرفته ۶,۲: تعریف متغیرها در داخل حلقه	۳۲۷
۶,۳ حلقه‌های تودرتو	۳۲۸
۶,۴ پردازش مقادیر نگهبانی	۳۳۱
نکته مدیریتی ۶,۲: حدود متقارن و نامتقارن	۳۳۷
نکته مدیریتی ۶,۴: شمارش تکرار	۳۳۸
مبحث پیشرفته ۶,۲: مشکل در حلقه ناقص	۳۳۹
مبحث پیشرفته ۶,۴: دستورات break و continue	۳۴۰
۶,۴ اعداد تصادفی و شبیه‌سازی برنامه‌ها	۳۴۲
مبحث پیشرفته ۶,۵: حلقه‌های تغییرناپذیر	۳۴۸
۶,۶ اشکالیابی	۳۴۹
۶,۷ نمونه دوره‌ای برای اشکالیابی	۳۵۴
نحوه اشکالیابی ۶,۲	۳۶۰
وقایع تصادفی ۶,۲: اولین باگ	۳۶۰

پیشگفتار مترجم

برنامه‌نویسی، شاخه‌ای از علم کامپیوتر است. با توجه به گسترش روز افزون علم و فناوری، شاهد پیدایش نرم‌افزارها و سخت‌افزارهای جدیدی با قابلیت‌های بسیار کاربردی و حساس می‌باشیم؛ نرم‌افزارهایی که از لحاظ امنیتی، مدیریتی، محاسباتی، شبیه‌سازی و ... سهم به‌سزایی در پیشبرد علم و دانش بشری ایفاء می‌کنند.

اما این برنامه‌ها را با کدام زبان‌های برنامه‌نویسی تولید می‌کنند و بهتر است که برنامه‌نویسی به کدام زبان‌ها را فرا بگیریم تا در آینده، برنامه‌نویس خبره‌ای باشیم؟

در سال‌های اخیر، مهمترین زبان‌های برنامه‌نویسی، JAVA، Visual C#، Python، و visual F# می‌باشند که هر کدام برای کار خاصی بکار می‌روند. در واقع انتخاب اینکه برنامه‌نویسی با کدام زبان را شروع کنید، به خودتان بستگی دارد، اینکه در چه حوزه‌ای تمایل به برنامه‌نویسی دارید؟ برای چه کاری برنامه‌نویسی می‌کنید؟ هدف شما از برنامه‌نویسی چیست؟

در حقیقت هر زبان برنامه‌نویسی دارای خصوصیات و ویژگی‌های منحصر بفرد است. مثلاً زبان C و ++C معمولاً برای ماشین‌های صنعتی بکار می‌روند، زبان فرترن برای انجام امور مهندسی و محاسباتی و همچنین زبان Visual Basic برای آموزش مبتدیان استفاده می‌شود.

درباره مؤلف و کتاب

کتاب حاضر، آموزش زبان برنامه‌نویسی جاوا می‌باشد که اثر دکتر کای هورستمن^۱، استاد دانشگاه سان جوز آمریکا بوده و مرجع کاملی برای آموزش جاوا در بسیاری از دانشگاه‌های مطرح است. مؤلف با توجه به مخاطب دانشجویان، کتاب را با زبانی ساده، شیوا و ابتدایی تألیف نموده است.

با توجه به کمبود منابع آموزشی برای زبان برنامه‌نویسی جاوا در ایران، بر آن شدیم تا با ترجمه ویرایش چهارم این کتاب، این خلاء را پر کنیم. اما چرا زبان جاوا را انتخاب کرده‌ایم؟ شاید بپرسید که چرا با وجود این همه تنوع زبان‌های برنامه‌نویسی، جاوا را مد نظر قرار داده‌ایم؟ با توجه به رتبه‌بندی سال‌های اخیر در سایت‌های اینترنتی، زبان جاوا در جهان از محبوبیت ویژه‌ای برخوردار شده است. در جدول زیر رتبه‌بندی زبان‌های برنامه‌نویسی را در سال‌های ۲۰۱۰ و ۲۰۱۱ مشاهده می‌کنید. چنانکه در این جدول آمده است، زبان جاوا علاوه بر رشد روز افزون، همچنان در صدر

^۱ Cay's Horstmann (University of San Jose – California, United States)

قرار دارد، چراکه برنامه‌های تولیدی آن را می‌توان در هر دستگاه و وسیله الکترونیکی (کلیدی، لمسی و غیر لمسی) اجرا نمود؛ در واقع این زبان، مستقل از سخت‌افزار است. شعار اصلی سازندگان جاوا، "یک بار بنویس، همه جا اجرا کن" است و مشاهده می‌کنیم که این شعار به واقعیت مبدل شده است. در حال حاضر اکثر برنامه‌های موبایل، PDAها، اپلت‌ها، برنامه‌های کاربردی، اینترنتی، صنعتی و ... توسط جاوا تولید و توسعه داده می‌شوند.

Delta in Position	Language	2011	2010	Status
↑	Java	19.043%	+0.99%	A
↓	C	16.162%	-1.90%	A
=	C++	9.225%	-0.48%	A
↑↑	C#	7.185%	+2.75%	A
↓	PHP	6.584%	-3.08%	A
↑	Python	4.931%	+0.73%	A
↓↓	(Visual) Basic	4.682%	-1.71%	A
↑↑↑	Objective-C	4.386%	+2.10%	A
↓	Perl	1.991%	-1.56%	A
=	JavaScript	1.513%	-0.96%	A
↑	Ruby	1.482%	-0.74%	A
↑↑↑↑↑↑↑	Lua	1.035%	+0.51%	A
↓↓↓	Delphi	1.034%	-1.68%	A
=	Assembly	0.967%	0.00%	A
↑↑↑↑↑↑↑	Lisp	0.934%	+0.45%	A
↑↑↑↑↑↑↑↑	Ada	0.768%	+0.41%	A
↓	Pascal	0.713%	+0.06%	A
↑↑↑	Transact-SQL	0.583%	+0.08%	B
=	Scheme	0.581%	0.00%	B
↓↓↓	Go	0.557%	-0.15%	A--

جدول رتبه بندی جهانی زبان‌های برنامه‌نویسی در سال‌های ۲۰۱۰ و ۲۰۱۱

در رابطه با فصل‌های کتاب

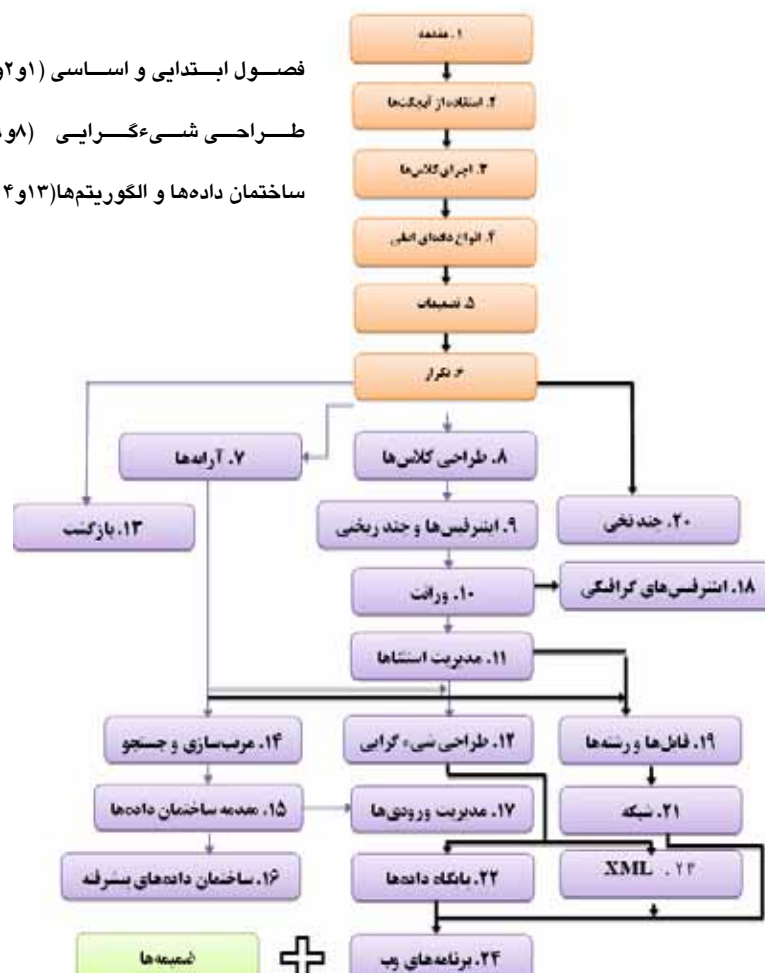
لازم به ذکر است که این کتاب در دو جلد ارائه شده است. با توجه به نمودار فصل‌ها مشاهده می‌کنید که در جلد اول، فصل‌های ۱ تا ۶ آمده‌اند، این فصل‌ها حاوی مطالب ابتدایی و اساسی برنامه‌نویسی جاوا هستند و یادگیری این مطالب برای برنامه‌نویسی به هر زبان دیگری لازم و ضروری است. همچنین توجه کنید که مطالب این ۶ فصل کاملاً با هم مرتبط بوده و باید به همان ترتیب مطالعه شود.

در جلد دوم، فصل‌های ۷ تا ۱۶ آمده‌اند، در این جلد، طراحی شیء‌گرایی، ساختمان داده‌ها و الگوریتم‌ها و ... آموزش داده شده‌اند. این مطالب، پیش‌نیازی برای دروس ساختمان داده‌ها و الگوریتم‌ها در دوره کارشناسی کامپیوتر است.

فصول ابتدایی و اساسی (۱ و ۲ و ۳ و ۴ و ۵ و ۶ و ۷)

طراحی شیء‌گرایی (۸ و ۹ و ۱۰ و ۱۱ و ۱۲)

ساختمان داده‌ها و الگوریتم‌ها (۱۳ و ۱۴ و ۱۵ و ۱۶ و ۱۷)



در مورد لوح فشرده و برنامه‌ها

در لوح فشرده همراه این کتاب، آخرین نسخه نرم‌افزارهای مورد نیاز یعنی Eclipse، Net Beans و همچنین آخرین نسخه^۱ JDK به همراه سورس برنامه‌های داخل کتاب ارائه شده است. به سه روش می‌توان برنامه‌ها را نوشت:

- بدون استفاده از محیط برنامه‌نویسی و با استفاده از notepad یا یک ویرایشگر متن
- با استفاده از برنامه Eclipse (نرم‌افزاری پرتابل، بدون نیاز به نصب)
- با استفاده از برنامه Net Beans (دارای اشارات و راهنمایی‌های سومند)

در صورتی که زمان یا فضای کافی برای نصب برنامه Net Beans ندارید، از برنامه Eclipse استفاده کنید، در غیر اینصورت برنامه Net Beans را نصب کنید. در این برنامه، راهنمایی‌های بسیار سودمندی در رابطه با کلاس‌ها، متدها، توابع و ... (در هنگام کد نویسی) ارائه می‌شود و برنامه‌نویسی در محیط آن راحت‌تر است.

توجه کنید که پیش از اقدام به نصب Net Beans یا استفاده از Eclipse، برنامه JDK را نصب کنید. بدون نصب این برنامه نمی‌توانید از آن‌ها استفاده کنید.

توجه کنید که در زبان جاوا تبدیلی وجود دارد که کدهای اجرایی توسط کامپایلر را به برنامه قابل اجرا تبدیل می‌کند، این مبدل، ماشین مجازی جاوا یا JVM نام دارد. بنابراین در هر جای کتاب با عبارتی مانند JVM یا ماشین مجازی مواجه شدید منظور همین مبدل است.

در انتها از خواهر گرانقدرم، زهره پوراحمد که در امر ویرایش و ترجمه کتاب کمک شایانی به بنده نمودند، نهایت تشکر را دارم. همچنین از جناب آقای مهندس حسین یعسوبی، مدیر انتشارات پندارپارس، به خاطر راهنمایی‌های سودمند، تشکر وافر خود را اعلام می‌دارم.

با توجه به اینکه هیچ مجموعه‌ای عاری از اشکال نمی‌باشد، لذا خواهشمند است در صورت یافتن مشکلات احتمالی، از طریق پست الکترونیک با بنده در تماس باشید. همچنین پذیرای نظرات، انتقادات و پیشنهادات سازنده شما عزیزان هستیم.

E-mail: apco_pourahmad@yahoo.com Weblog: www.apcomputer.blogfa.com

انور پوراحمد - بهار ۱۳۹۰

^۱Java Development Kit

فصل ۱

مقدمه

اهداف فصل

- درک قابلیت‌های برنامه نویسی
- آموختن پیرامون معماری کامپیوتر
- آموختن پیرامون زبان ماشین و برنامه نویسی به زبان سطح بالا
- آشنایی با کامپیوتر و کامپایلرها
- کامپایل کردن و اجرای اولین برنامه جاوا
- تشخیص خطاهای نحوی و منطقی

هدف این فصل آشنایی با مفهوم برنامه نویسی است و در رابطه با معماری کامپیوتر و بحث در رابطه با تفاوت میان زبان ماشین و زبان‌های سطح بالای برنامه نویسی است. نهایتاً چگونگی کامپایل، اجرای اولین برنامه جاوا و تشخیص خطاهایی که ممکن است هنگام کامپایل یا اجرای برنامه رخ دهد را مشاهده خواهید کرد.

برنامه نویسی چیست؟

احتمالاً برای کارهای شخصی یا سرگرمی از کامپیوتر استفاده می‌کنید. بیشتر مردم برای وظایف روزانه مانند بررسی حساب‌ها، ارسال ایمیل و ... از کامپیوتر استفاده می‌کنند. کامپیوتر برای اینگونه وظایف مناسب است. کامپیوتر را برای کارهای تکراری روزمره مانند جمع کردن اعداد بزرگ یا جایگذاری کلمات در یک صفحه، می‌توان بکاربرد. همچنین کامپیوترها بعنوان ماشین یا ابزار سرگرم کننده مورد استفاده قرار می‌گیرند، زیرا آنها می‌توانند ترکیبی از صدا و تصویر را نمایش داده و کاربران را مشغول و سرگرم نمایند.

چند منظوره بودن کامپیوترها پدیده‌ای کاملاً شگفت‌انگیز است، کامپیوتر نامه‌ها را چاپ، بازی‌ها را اجرا و در انجام کارهای روزمره به شما کمک می‌کند.

کامپیوتر برای اجرای کارهای متفاوت باید برنامه ریزی شود. کارهای متفاوت به برنامه‌های مختلفی نیاز دارند.

برای دست‌یابی به این هدف، باید کامپیوتر برای اجرای هر وظیفه برنامه‌ریزی شود. کامپیوتر یک نوع ماشین محسوب می‌شود که داده‌ها را ذخیره می‌کند (اعداد، کلمات، تصاویر)، با دستگاه‌ها ارتباط برقرار می‌کند (صفحه نمایش، سیستم پخش صدا، چاپگر) و برنامه‌ها را اجرا می‌کند. برنامه‌ها سلسله‌مراتبی از دستورالعمل‌ها و تصمیمات هستند که برای رسیدن به هدف و کار مشخص، توسط کامپیوتر انجام می‌شوند.

برنامه کامپیوتری عملیات پایه‌ای متوالی را به ترتیب و سریع اجرا می‌کند.

برنامه‌های کامپیوتری امروزی خیلی پیچیده هستند، در حالیکه همه آنها ترکیبی از دستورالعمل‌های ابتدایی هستند.

عملیاتی که کامپیوتر انجام می‌دهد، می‌تواند یکی از موارد زیر باشد:

- قراردادن نقطه قرمز در ناحیه‌ای از صفحه نمایش
- چاپ نامه بوسیله چاپگر
- بدست آوردن موقعیت برنامه در حافظه
- جمع نمودن دو عدد با همدیگر

برنامه به کامپیوتر دستور می‌دهد که در زمان مورد نظر کار مشخصی را انجام دهد. تعداد عملیاتی که هنگام اجرای برنامه انجام می‌شود، بسیار زیاد است و کامپیوتر آنها را در سرعت بالایی اجرا می‌کند. در حقیقت کامپیوترها هوش ندارند و کارهایی که انجام می‌دهند، ترتیبی از دستورالعمل‌ها است.

برنامه کامپیوتری شامل دستورالعمل‌های متوالی برای دستیابی به هدف برنامه است.

دانستن روش‌ها و اصول برنامه‌نویسی برای یک کاربر عادی کامپیوتر لازم نیست، این مشابه وقتی است که بدون آگاهی از نحوه کار موتور و قطعات ماشین، رانندگی می‌کنید. وقتی یک نامه را با برنامه واژه پرداز می‌نویسید، برنامه واژه پرداز توسط شرکتی برای این کار طراحی و تولید شده است. هدف ابتدایی این کتاب آموزش چگونگی طراحی و اجرای برنامه‌های کامپیوتری است، همچنین چگونگی فرمول‌بندی دستورالعمل‌ها را یاد خواهید گرفت.

به یاد داشته باشید که برنامه‌نویسی یک بازی کامپیوتری پیچیده یا یک برنامه واژه پرداز متن، به تیمی متشکل از تعداد زیادی برنامه‌نویس ماهر، گرافیسیت و افراد حرفه‌ای دیگر نیازمند است. اولین تلاش شما برای شروع برنامه‌نویسی، زمان زیادی به طول می‌انجامد. بطور کلی در این کتاب، نحوه صحیح و اصولی برنامه‌نویسی به روشی پایه‌ای و اساسی را خواهید آموخت، اما نباید انتظار ایجاد فوری نرم افزارهای حرفه‌ای را داشته باشید. یک برنامه دانشگاهی در علوم کامپیوتر یا مهندسی نرم‌افزار، چهار سال طول می‌کشد تا کامل شود؛ در این کتاب قصد داریم آموزش‌های مقدماتی برای نوشتن چنین برنامه‌هایی را ارائه دهیم. بسیاری از دانشجویان از کار با برنامه‌های مختلف کامپیوتری به هیجان و نشاط می‌آیند، زیرا کاری را که ساعت‌ها برای آن تلاش می‌کنیم، کامپیوتر در مدت زمان اندکی انجام می‌دهد.

خودآزمایی

۱. برای پخش CD موزیک در کامپیوتر چه چیزی لازم است؟
۲. چرا CD player از کامپیوتر قابلیت شخصی سازی کمتری دارد؟
۳. آیا می‌توان برنامه کامپیوتری را ابتکاری برای اجرای وظایف و کارها بصورت سریع‌تر و دقیق‌تر دانست؟

۲.۱ ساختمان کامپیوتر (آناتومی کامپیوتر)

برای فهمیدن روند برنامه نویسی، باید در رابطه با Case کامپیوتر آشنایی مقدماتی داشته باشید. در این بخش، یک کامپیوتر شخصی را بررسی خواهیم کرد. کامپیوترهای بزرگ (ابر کامپیوترها) بزرگتر و سریعتر هستند اما آن‌ها در مقابل کامپیوترهای شخصی، اساساً طراحی یکسانی دارند.

در قلب کامپیوتر، واحد پردازنده مرکزی (CPU) واقع است (شکل ۱-۱ را مشاهده کنید). CPU شامل یک تراشه (مدار مجتمع) یا تعداد کمی از تراشه‌ها است. تراشه کامپیوتری ترکیبی از پلاستیک یا محفظه فلزی، رابط‌های فلزی و سیم‌کشی داخلی اصلی ساخته شده از ماده سیلیکون است. تراشه CPU، دارای سیم‌کشی داخلی فوق‌العاده پیچیده‌ای است. برای مثال، تراشه پنتیوم ۴ (یک CPU رایج برای کامپیوترهای خانگی در سال‌های ۲۰۰۲ الی ۲۰۰۸) شامل بیش از ۵۰ میلیون عنصر بنیادی است که ترانزیستور نامیده می‌شوند. اینها عناصری هستند که سیگنال‌های الکتریکی را برای کنترل سیگنال‌های الکتریکی دیگر بکار برده و باعث می‌شود محاسبات خودکار (اتوماتیک) امکان‌پذیر شود. CPU دستورالعمل‌های برنامه را تعیین و اجرا می‌کند، عملیاتی مانند جمع، تفریق، ضرب و تقسیم را به صورت اتوماتیک انجام می‌دهد و داده‌ها را از محل ذخیره (دیسک سخت) و وسایل ورودی یا خروجی آورده و اطلاعات را برمی‌گرداند.

در قلب کامپیوتر، واحد پردازنده مرکزی (CPU) قرار دارد .



شکل ۱-۱ واحد پردازنده مرکزی

کامپیوتر داده و برنامه‌ها را در حافظه نگهداری می‌کند. دو نوع قطعه ذخیره اطلاعات وجود دارد: حافظه اصلی و حافظه ثانوی. حافظه اصلی، معمولاً حافظه با دسترسی تصادفی (RAM)، سریع اما گران قیمت است، و وقتی کامپیوتر خاموش شود تمام داده‌های روی آن، از بین می‌روند. حافظه ثانوی، معمولاً دیسک سخت (Hard Disk)، دستگاه ذخیره سازی ارزان‌تر است که حتی با قطع برق، اطلاعات را نگهداری می‌کند. دیسک سخت شامل دیسک گردان، که با ماده مغناطیسی پوشیده شده- است و هد‌های خواندن و نوشتن، که می‌تواند طرح‌های مختلف خطوط مغناطیسی روی دیسک را بخواند و تغییر دهد، می‌باشد. این روند ضبط و پخش اساساً با روندی که در نوارهای صوتی و تصویری استفاده می‌شود، یکسان است.

داده‌ها و برنامه‌ها در حافظه اصلی و حافظه ثانوی (مانند دیسک سخت) ذخیره می‌شوند.



شکل ۲-۱ مدلی از حافظه با تراشه‌های حافظه روی آن



شکل ۳-۱ دیسک سخت

در حالی که کامپیوترها بوسیله شبکه‌ها به همدیگر وصل می‌شوند، برخی از آن‌ها نسبت به برخی دیگر کامل‌تر هستند. کامپیوترهای خانگی معمولاً با dialup یا اتصال با پهنای باند بالا به اینترنت وصل می‌شوند. کامپیوترها می‌توانند اطلاعات را بوسیله ارتباط شبکه از کامپیوتر مرکزی که اصطلاحاً سرور نامیده می‌شود، دریافت کنند و یا اطلاعات را به آن‌ها انتقال دهند. برای کاربر مبتدی که از کامپیوتری بر روی شبکه استفاده می‌کند، تشخیص اینکه کدامیک از اطلاعات روی کامپیوتر وی است یا روی سرور(کامپیوتری که روی اینترنت یا شبکه فایل‌ها را با شما به اشتراک می‌گذارد)، کمی سخت خواهد بود.

بیشتر کامپیوترها دستگاه‌های ذخیره اطلاعات قابل حمل دارند که از طریق آن‌ها می‌توان به داده‌ها و برنامه‌ها بر روی رسانه‌هایی مانند فلاپی‌ها، نوارها یا دیسک‌های فشرده (DVD,CD) دسترسی پیدا کرد.

کاربر کامپیوتر به دستگاه‌های جانبی دیگر نیاز دارد. کامپیوتر اطلاعات را به وسیله صفحه نمایش، بلندگوها و چاپگرها به کاربر انتقال می‌دهد. کاربر می‌تواند اطلاعات و دستورات را با استفاده از صفحه کلید یا دستگاه اشاره‌گر مانند ماوس به کامپیوتر انتقال دهد.



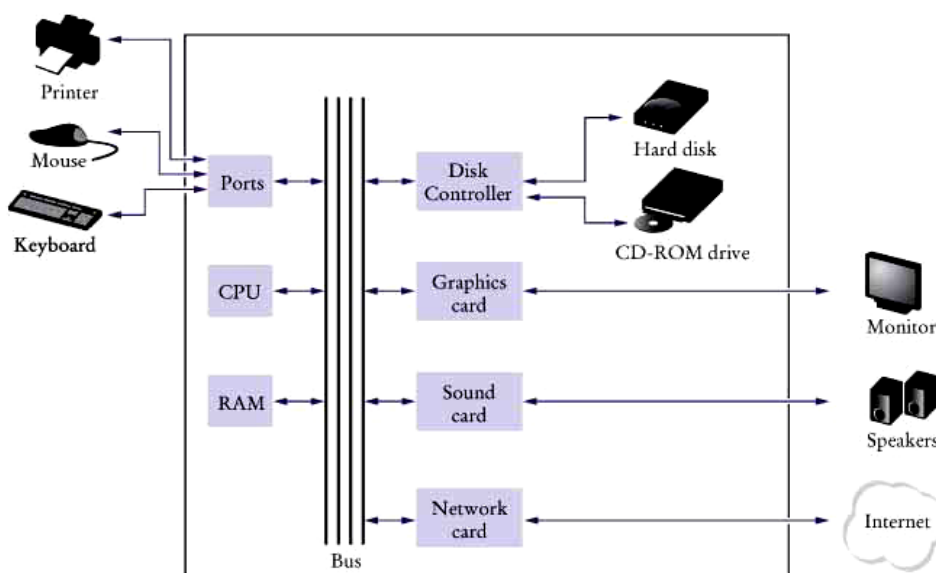
شکل ۱-۴ مادربرد

CPU، رم، دیسک سخت و دستگاه‌های دیگر به وسیله دستگاهی متشکل از خطوط الکتریکی به نام باس (bus) به هم وصل هستند. شکل ۱-۴ یک مادربرد را نشان می‌دهد که شامل مکان‌های CPU، رم و اتصال دهنده دستگاه‌های جانبی است.

شکل ۱-۵ مروری بر طرح کلی معماری کامپیوتر ارائه می‌دهد. دستورات برنامه‌ها و داده‌ها (مانند متن، اعداد، صدا یا تصویر) بر روی دیسک سخت، سی‌دی یا شبکه ذخیره می‌شوند. وقتی برنامه شروع به کار می‌کند، به داخل حافظه برای خوانده شدن بوسیله CPU می‌رود. بوسیله این دستورات، CPU داده‌ها را می‌خواند، آنها را تغییر می‌دهد و آنها را به حافظه اصلی یا حافظه ثانوی برمی‌گرداند. برخی از برنامه‌ها برای ارتباط و کنترل صفحه نمایش، بلندگوها و... طراحی شده‌اند، به دلیل اینکه این

اعمال بیش از چندین بار و در سرعت بالا اتفاق می‌افتند، کاربران تصاویر و صدا را درک می‌کنند. به همین روال، CPU دستورات را به چاپگر می‌فرستد برای چاپ بر روی کاغذ با طرحی از نقاط فاصله-دار بادقت، که انسان آن را مانند متنی از حروف و عکس‌ها تشخیص می‌دهد، همچنین برخی از دستورات ورودی کاربر را از صفحه کلید و ماوس می‌خواند و آن‌ها را پردازش می‌کند. برنامه‌ها ماهیت این ورودی‌ها را تحلیل کرده و سپس دستور مناسب بعدی را اجرا می‌کنند.

CPU دستورات ماشین را از حافظه می‌خواند. دستورات مستقیماً با حافظه اصلی، حافظه ثانوی و دستگاه‌های جانبی دیگر ارتباط برقرار می‌کنند.



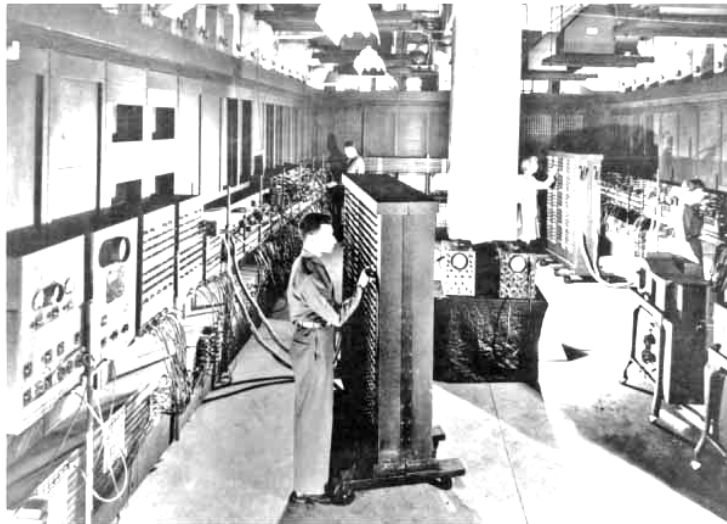
شکل ۱-۵ نمودار کلی معماری کامپیوتر

خودآزمایی

۴. برنامه هنگامی که اجرا نشده است، کجا ذخیره می‌شود؟
۵. کدام قسمت از کامپیوتر عملیاتی مانند جمع و ضرب را بصورت خودکار انجام می‌دهد؟

وقایع تصادفی ۱,۱: ENIAC و آغاز محاسبات (پیدایش کامپیوترها)

انیاک (ENIAC) یکی از بزرگترین کامپیوترهای الکترونیکی (اولین کامپیوتر الکترونیکی قابل استفاده بود که بوسیله J. Presper Eckert و John Mauchly در دانشگاه پنسیلوانیا^۱ طراحی شد و در سال ۱۹۴۶ تکمیل گردید. بجای ترانزیستور، که تا آن موقع اختراع نشده بود و دو سال بعد ساخته شد، انیاک شامل حدود 18,000 لامپ خلاء در تعداد زیادی جایگاه در یک اتاق بزرگ بود. لامپهای خلاء به میزان چندین لامپ در روز می سوختند. نگهداری همراه یک گاری پر از لامپ دائماً دور می زد و لامپ-های معیوب را جایگزین می کرد. کامپیوتر بوسیله سیمهای متصل کننده روی پانل (صفحات و مدارات) برنامه ریزی شده بود. برای این که کامپیوتر کار دیگری انجام می داد، بایستی سیمها دوباره به روش دیگری بسته می شدند.



انیاک

نیروی دریایی آمریکا بر روی انیاک کار می کرد. نیروی دریایی آمریکا که علاقه مند به محاسبه جدول حرکت پرتابه (موشک) در مقاومت باد، با سرعت اولیه و شرایط جوی بود، از انیاک استفاده می کرد. برای محاسبه مسیر گلوله، باید یک راه حل از معادله دیفرانسیلی معین پیدا می شد که از آن زمان انتگرال گیری عددی نامیده شد. قبل از این که ماشین هایی مانند انیاک توسعه داده شوند، انسان این نوع کارها را انجام می داد و تا سال ۱۹۵۰ این مردم بودند که کار کامپیوترها را انجام می دادند. انیاک بعداً برای اهداف دیگری، مانند سرشماری مردم آمریکا استفاده شد.

¹ University of Pennsylvania

۱,۳ تبدیل برنامه‌های قابل فهم انسان به زبان ماشین

دستورات قابل فهم برای کامپیوتر بسیار ابتدایی هستند و پردازشگر تنها دستورات ماشین را اجرا می‌کند. CPUهای شرکت‌های مختلف، مانند اینتل یا سان، دستورات ماشین متفاوتی دارند. برنامه‌های جاوا بدون نیاز به اصلاح و تغییر بر روی CPUهای متفاوت اجرا می‌شوند، برنامه‌های جاوا شامل دستورات ماشین هستند که اصطلاحاً ماشین مجازی جاوا یا JVM نامیده می‌شود، این برنامه برای اجرا شدن برنامه کدها را برای CPU شبیه سازی می‌کند. تفاوت میان دستورات ماشین واقعی یا مجازی مهم نیست، فقط کافی است بدانید که دستورات ماشین بسیار ساده و بصورت رمزی هستند (مانند اعداد ذخیره شده در حافظه) و خیلی سریع اجرا می‌شوند.

به طور کلی، زبان ماشین به نوع CPU وابسته است. با این وجود، دستورات ماشین مجازی جاوا (JVM) می‌توانند روی چندین CPU اجرا شوند.

نمونه‌ای از دستورات ترتیبی ماشین عبارت است از:

۱. بارگذاری محتویات ۴۰ مکان حافظه
۲. اختصاص ۱۰۰ مقدار
۳. اگر اولین مقدار از دومین مقدار بزرگتر بود، از دستوراتی که در محل ۲۴۰ حافظه ذخیره شده است دوباره ادامه بده.

در حقیقت، دستورات ماشین مانند اعداد برای اینکه بتوانند در حافظه ذخیره شوند، بصورت رمزی در می‌آیند. ماشین مجازی جاوا، این دستورات ترتیبی را به اعداد رمزی تبدیل می‌کند.

۴۰ ۲۱

۱۰۰ ۱۶

۲۴۰ ۱۶۳

وقتی که ماشین مجازی جاوا این ترتیب از اعداد را می‌خواند، رمز آنها گشوده می‌شود و ترتیبی پیوسته از فرمان‌ها را اجرا می‌کند.

چگونه می‌توان با فرمان‌های متوالی با کامپیوتر ارتباط برقرار کرد؟ بیشتر متدها بصورت مستقیم و خطی اعداد واقعی را داخل حافظه کامپیوتر جای می‌دهند. با این وجود برنامه طولانی ترکیبی از هزاران فرمان منحصر بفرد است و جست‌وجوی کدهای خطی و فرمان‌ها در حافظه خسته کننده و احتمالاً همراه با خطا می‌باشد. چنانکه که قبلاً اشاره شد، کامپیوترها در فعالیت‌های خودکار، معمولاً

خطاهایی دارند، به همین دلیل برنامه نویسان کامپیوتر در پروسه برنامه نویسی نمی‌توانند بطور کامل برنامه را در اختیار خود بگیرند.

به دلیل اینکه دستورات ماشین به صورت عددی کد گذاری می‌شوند، نوشتن برنامه‌ها به زبان ماشین برای انسان دشوار است.

در اواسط سال ۱۹۵۰، زبان‌های برنامه نویسی سطح بالا ظاهر شدند. در این زبان‌ها، برنامه نویس، هدف از انجام پروژه برنامه نویسی را بیان می‌کند و برنامه را می‌نویسد سپس برنامه ویژه کامپیوتری به نام کامپایلر آن‌ها را به زبان ماشین تبدیل می‌کند.

زبان‌های سطح بالا به شما اجازه می‌دهند برنامه‌ها را در سطح مفهومی‌تری از زبان ماشین بنویسید.

برای مثال در جاوا (زبان سطح بالای برنامه نویسی که شما در این کتاب از آن استفاده خواهید کرد) می‌توانید دستور زیر را بنویسید:

```
if (intRate > 100)
```

```
System.out.println("Interest rate error");
```

این یعنی: اگر میزان ورودی بیش از ۱۰۰ بود، پیغام خطایی را نمایش بده. این کار برنامه کامپایلر است که به ترتیب حروف if (intRate > 100) نگاه کند و آن را به:

```
21      40      16      100      163      240      ...
```

تبدیل کند. کامپایلرها برنامه‌های کاملاً پیچیده‌ای هستند، آن‌ها دستورات منطقی، مانند دستور if را تبدیل به رشته‌های عددی می‌کنند و برنامه را اجرا می‌کنند. کامپایلرها محل‌هایی از حافظه را به متغیرها اختصاص می‌دهند. اگر تصمیم دارید متخصص کامپیوتر شوید، باید در مطالعات خود چیزهای زیادی در مورد تکنیک‌های نوشتن کامپایلر را بگنجانید.

کامپایلر برنامه‌های نوشته شده به زبان سطح بالا را به زبان ماشین تبدیل می‌کند.

خودآزمایی

۶. کد ماشین جاوا برای دستور بارگذاری محتویات موقعیت ۱۰۰ حافظه، چیست؟
۷. آیا یک شخص که از کامپیوتر برای کار اداری استفاده می‌کند، به اجرای کامپایلر نیاز دارد؟

۱.۴ زبان برنامه‌نویسی جاوا

در سال ۱۹۹۱، گروهی به رهبری جیمز گاسلینگ (James Gosling) و پاتریک ناوتن (Patrick Naughton) در شرکت سان زبان برنامه‌نویسی طراحی کردند که از آن در مصرف‌کننده‌هایی مانند تلویزیون هوشمند استفاده می‌شد و آن را Green نامیدند. زبان طراحی شده ساده و همراه با معماری خاصی بود، به طوری که توانایی اجرا بر روی سخت‌افزارهای مختلف را داشته باشد.

جاوا در اصل برای برنامه‌نویسی دستگاه‌های مصرفی طراحی شد، اما اولین بار در استفاده برای نوشتن اپلت‌های (applet) اینترنت موفقیت یافت.

گاسلینگ در سال ۱۹۹۴ تیم واقعیش را با شعار: "ما می‌توانیم یک مرورگر واقعی بسازیم، کمبودهایی که در مسیر شبکه به آن‌ها نیاز داریم: معماری خاص، زمان انجام، اعتبار و امنیت در برنامه‌نویسی"، معرفی نمود. جاوا در اجتماعی از علاقه‌مندان برنامه‌نویسی در سال ۱۹۹۵ در نمایشگاه SunWorld معرفی شد.

از آن پس، جاوا با سرعتی خارق‌العاده توسعه یافت. برنامه‌نویس‌ها زبان جاوا را پذیرفتند زیرا جاوا از نزدیک‌ترین رقیبش یعنی ++C ساده‌تر است. علاوه بر این، جاوا کتابخانه‌ای غنی دارد که امکان نوشتن برنامه‌های قابل حمل را فراهم کرده است.

جاوا مطمئن و قابل حمل طراحی شده بود و برای استفاده کاربران اینترنت و دانشجویان مفید بود.

جاوا برای اینترنت طراحی شده بود. دو ویژگی جالب توجهی که جاوا برای مبتدیان دارد ایمنی و قابل حمل بودن است. اگر یک صفحه وب را که حاوی کدهای جاوا (به اصطلاح اپلت‌ها) است را بازدید کنید، کدها به طور خودکار اجرا می‌شوند. اگر یک اپلت مضراتی مانند صدمه زدن به داده‌ها یا خواندن اطلاعات شخصی داشت، شما هر زمان که وب را مرور می‌کردید در معرض خطر واقعی بودید - یک برنامه‌نویس بی احتیاط ممکن است صفحه وبی شامل کدهای مخرب را طراحی کند که به محض بازدید صفحه، کدهای مخرب در سیستم شما نفوذ کند. زبان جاوا شامل مجموعه‌ای از ویژگی‌های امنیتی است که تضمین می‌کند اپلت‌های مضر نمی‌توانند روی کامپیوتر شما اجرا شوند. ماشین جاوا بسیاری از خطاهای مبتدیان را می‌یابد و به دقت گزارشی از آنها را می‌دهد. (در مقابل، بیشتر خطاهای مبتدیان در ساخت برنامه‌های به زبان ++C صرفاً به راه‌های گیج‌کننده و تصادفی می‌انجامد). مزیت دیگر جاوا قابلیت حمل برنامه‌های آن است. اکثر برنامه‌های جاوا، بدون تغییر، بر روی سیستم عامل‌های ویندوز، یونیکس، لینوکس و مکینتاش اجرا می‌شوند، این ویژگی کاربرد مفید

اپلت‌ها را نشان می‌دهد. وقتی یک صفحه وب را بازدید می‌کنید، سرور وب سایتی که محتویات صفحه را بارگزاری می‌کند نمی‌داند چه کامپیوتری بر روی اینترنت، جستجو می‌کند، این قابلیت حمل کدهایی را که توسط کامپایلر جاوا تولید شده‌اند را نشان می‌دهد. کدهای قابل حمل توسط ماشین جاوا اجرا می‌شوند، این قابلیت مفیدی برای دانشجویان است؛ به همین دلیل دانشجویان مجبور به نوشتن برنامه‌های متفاوت برای سیستم عامل‌های مختلف نیستند.

امروزه، جاوا یکی از زبان‌های مهم برنامه نویسی برای اهداف شخصی و همچنین آموزش علوم کامپیوتر تبدیل شده است. با وجود این که جاوا زبان مناسبی برای مبتدیان است، اما به سه دلیل کامل نیست:

زیرا جاوا به صورت اختصاصی برای دانشجویان طراحی نشده است، ایده خاصی برای ساده کردن طراحی و نوشتن برنامه‌ها وجود ندارد. حداقل سیستم مشخصی برای نوشتن برنامه های جاوا لازم است. این برای دانشجویان یک مشکل است اما برای برنامه نویسان حرفه ای مشکلی بوجود نمی‌آورد. همزمان که برنامه نویسی در جاوا را یاد می‌گیرید، بارها برای شما سؤالاتی پیش می‌آید که پاسخ آنها را با توضیحات پایه‌ای و جزئیات کامل در فصول بعدی خواهید آموخت.

زبان جاوا از زمان پیدایش بارها بازبینی و تکمیل شده است، در جدول ۱-۱ می‌توانید این بازبینی‌ها را ببینید. در این کتاب، فرض ما بر این است که از نسخه ۵ یا بالاتر آن استفاده می‌کنید.

نسخه	سال	ویژگی‌های جدید
۱,۰	۱۹۹۶	-----
۱,۱	۱۹۹۷	کلاس‌های داخلی
۱,۲	۱۹۹۸	-----
۱,۳	۲۰۰۰	افزایش کارایی
۱,۴	۲۰۰۲	اعلانات ، XML1
۵	۲۰۰۴	کلاس‌های عمومی ، اضافه شدن حلقه for ، auto-boxing ، شمارش
۶	۲۰۰۶	بهبود و توسعه کتابخانه

جدول ۱-۱ نسخه‌های جاوا

در نهایت باید بدانید که تنها در یک ترم نمی‌توان جاوا را به خوبی آموخت. زبان جاوا به خودی خود نسبتاً ساده است، اما جاوا مجموعه‌ای گسترده از بسته‌های کتابخانه‌ای دارد که برای نوشتن برنامه‌های مفید و قوی لازم است.

رابط کاربری برای رمز نگاری، شبکه، صدا، ذخیره اطلاعات و دیگر اهداف استفاده می‌شود. حتی برنامه نویسان ماهر جاوا امیدی به دانستن تمامی امکانات جاوا ندارند، آن‌ها فقط برای پروژه‌های مشخصی از آن‌ها استفاده می‌کنند.

جاوا کتابخانه بسیار بزرگی دارد. بر روی قسمت‌هایی از کتابخانه‌ها که برای پروژه‌های برنامه نویسی خود به آن نیاز دارید، تمرکز کنید.

با استفاده از این کتاب، باید انتظار یادگیری زبان جاوا و اکثر بسته‌های مهم آن را در حد مطلوبی داشته باشید. به خاطر داشته باشید هدف اصلی این کتاب حفظ جزئیات جاوا نیست، بلکه آموزش چگونگی تفکر در برنامه نویسی است.

خودآزمایی

۸. دو عدد از مهمترین مزایای جاوا را نام ببرید؟
۹. یادگیری کل کتابخانه‌های جاوا چه مدت طول می‌کشد؟

۱.۵ آشنایی با کامپیوتر

شاید خواندن این کتاب اولین دوره یادگیری برنامه‌نویسی شما باشد و شاید کار کردن روی سیستم‌های کامپیوتری برایتان جدید باشد. مقداری از زمان خود را صرف آشنایی با کامپیوتر کنید. از آنجا که سیستم‌های کامپیوتری متفاوت هستند، در این کتاب تنها می‌توانیم طرح کلی از مراحل برنامه نویسی به شما نشان دهیم. استفاده از سیستم کامپیوتری جدیدی که با آن آشنایی کمی دارید، کمی سخت است، به خصوص اگر راهنمایی همراهتان نباشد، برای این منظور در دوره‌های آموزشی که در دانشگاه یا آموزشگاه‌ها تشکیل می‌شود، شرکت کنید.

کمی از وقت خود را صرف آشنایی با سیستم‌های کامپیوتری و کامپایلر جاوا کنید.

مرحله ۱. ورود به سیستم

اگر از کامپیوتر خانگی استفاده می‌کنید، نیازی به نگرانی در مورد این مرحله نیست. کامپیوترهای آزمایشگاه، معمولاً در دسترس همه نیستند. شما احتمالاً به شماره کاربری و رمز عبور برای دسترسی به این گونه سیستم‌ها نیاز دارید.

مرحله ۲. تعیین محل کامپایلر جاوا

سیستم‌های کامپیوتری در این زمینه بسیار متفاوت هستند. در برخی از سیستم‌ها باید برنامه‌ای را باز کنید (به شکل ۱-۷ نگاه کنید) و دستوراتی را برای راه‌اندازی کامپایلر تایپ کنید. سیستم‌های دیگر