

**برنامه نویسی**

# نگاه متفاوت از زاویه‌های مختلف

پارسا ستوده‌نیا

همان طور که می‌دانیم، برنامه‌نویسی شاخه‌های گوناگون و زمینه‌های متفاوتی دارد. هر کدام از شاخه‌های اصلی نیز ممکن است خود به چند زیرشاخه تقسیم‌بندی شود. هر برنامه‌نویس در هر زمینه‌ای که فعالیت می‌کند باید به تمامی اهداف، امکانات و موانع کار خود آگاه باشد تا بتواند در نهایت محصولی کارا و مفید عرضه کند.

از مهم‌ترین و اصلی‌ترین شاخه‌های برنامه‌نویسی می‌توان به برنامه‌نویسی تحت شبکه، تحت وب، برنامه‌های مستقل رایانه‌های شخصی و برنامه‌نویسی تلفن‌های همراه اشاره کرد. هر کدام از این زمینه‌ها با داشتن فاکتورهای مهم و مورد نیاز مربوط به خود توجه خاص خود را طلب می‌کند. در ادامه سعی داریم نکاتی را که در هر زمینه بیشتر باید به آنها توجه داشت، به طور اجمالی بررسی و این زمینه‌ها را با هم مقایسه کنیم.

**برنامه‌های مستقل کاربردی**

یک دسته از برنامه‌ها هستند که

به صورت یک برنامه مستقل قابل اجرا روی یک رایانه عرضه می‌شوند.

بخش عمده‌ای از برنامه‌های رایانه‌ای از این دسته هستند. این گونه برنامه‌ها، بدون هیچ نوع نیاز به

ارتباط، به طور مستقل از سایر برنامه‌ها می‌توانند در رایانه شخصی کاربر اجرا شده و به نیازهایی پاسخ دهد. هنگام طراحی چنین برنامه‌هایی، باید سیستم‌عامل مورد نظر و کامپایلر سازگار با آن سیستم‌عامل را که بتواند زبان مورد استفاده ما را به برنامه اجرایی تبدیل کند، انتخاب کنیم.
معمولا برنامه‌نویس‌ها، سیستم‌عملی را انتخاب می‌کنند که در بین کاربران محبوب‌تر بوده و استفاده بیشتری می‌شود. زیرا به این ترتیب به نیاز عده بیشتری پاسخ داده خواهد شد. در حال حاضر سیستم‌عامل ویندوز در میان سایر سیستم‌عامل‌ها از محبوبیت بیشتری برخوردار بوده و کاربران بیشتری را به خود جلب کرده است.

یکی از مواردی که برنامه‌نویس باید به آن توجه کند میزان مصرف حافظه و دراختیار گرفتن پردازنده و دیگر منابع سیستم است. هرچه برنامه‌ها کمتر به منابع سیستم نیاز داشته باشند و آنها را کمتر در اختیار بگیرند و در عین حال بتوانند به همه نیازهای پیش‌بینی شده پاسخ بدهند، بهتر خواهند بود.

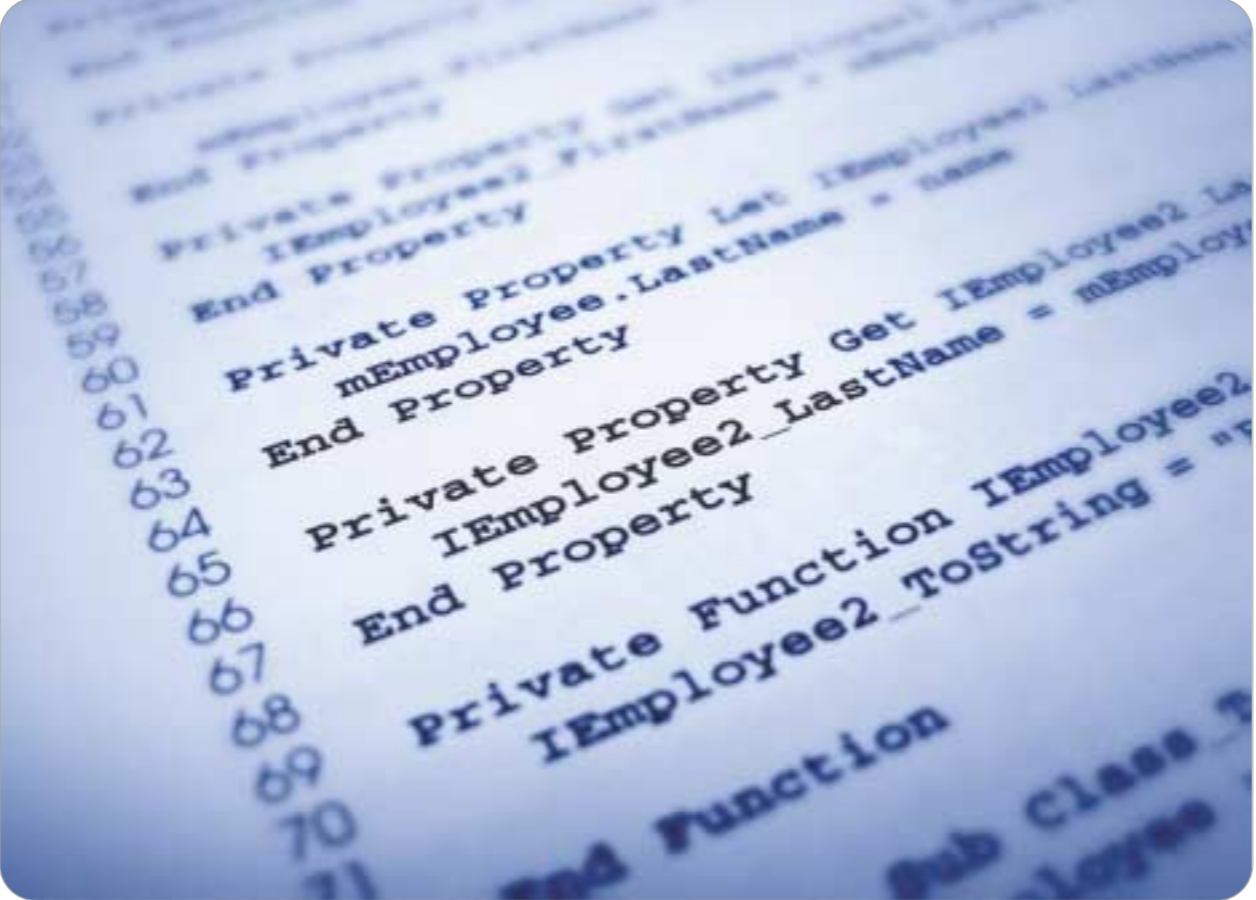
برنامه‌نویس باید توجه داشته باشد که در برنامه خود حتی‌الامکان کمتر از منابع سخت‌افزاری و نرم‌افزاری سیستم استفاده کند و در صورت در اختیار گرفتن آنها، بلافاصله بعد از استفاده و یا در زمانی که آن را نیاز

ندارد، رها کرده تا دیگر برنامه‌ها نیز بتوانند در صورت نیاز از آن بهره‌مند شوند. یک راه برای رعایت این مساله استفاده از حافظه به صورت دینامیک یا پویا است. زمانی که حافظه‌ای به صورت استاتیک به برنامه‌ای اختصاص داده می‌شود، در تمام طول مدت اجرای برنامه در اختیار آن خواهد بود و رها نخواهد شد. اما بسیاری از زبان‌های برنامه‌نویسی این امکان را به ما می‌دهند که هر میزان حافظه نیاز داریم در زمان اجرای برنامه از سیستم طلب کنیم و زمانی که کارمان با آن تمام شد، آن را آزاد کنیم. به این ترتیب برنامه‌کمترین میزان ممکن از حافظه را در طول اجرای خود مصرف خواهد کرد.

از سوی دیگر باید به سرعت اجرای برنامه توجه زیادی شود. ممکن است در صورت اشغال کمی حافظه بیشتر، کاربر را متوجه این امر نسازد، اما کندی برنامه و کم بودن سرعت از دید افراد پنهان نخواهد ماند. بنابراین به طور مثال اگر قرار است در طول اجرای برنامه هزاران بار حافظه گرفته و رها شود، بهتر است یک بار این کار به صورت استاتیک انجام شود تا مجبور نباشیم یک سری دستورالعمل را هزاران بار اجرا کنیم.

مساله دیگری که باید به آن توجه شود، واسط گرافیکی برنامه است. این که ظاهر برنامه زیبا باشد و طراحی آن کاربر پسند بوده و کار با آن راحت باشد از مهم‌ترین خصوصیات یک برنامه است.

اما باید توجه داشت که استفاده نابجا و بیش از حد



**برنامه‌نویسی تحت وب**

سرعت اجرای برنامه و سرعت پاسخگویی به کاربر مساله‌ای است که در همه زمینه‌های برنامه‌نویسی باید به آن توجه داشت. در برنامه‌نویسی تحت وب، سرعت پاسخگویی مستقیما با چند عامل در ارتباط است. کد برنامه و روش‌های به کار رفته در آنها همان‌طور که در پلوری طراحی کنیم که کاربر بتواند طریقه انجام بسیاری از کارها در برنامه را با استفاده از ظاهر برنامه حدس بزنند.

و انصراف از استفاده و مشاهده برنامه شما ترغیب کند. در طراحی برنامه‌های تحت وب باید به دو مساله دیگر نیز توجه داشت. یکی این که کاربران ممکن است با نمایشگرهایی با وضوح و ابعاد مختلف به مشاهده سایتی بپردازند. دیگر آن که به دلیل وجود سیستم عامل‌های گوناگون و همچنین متفاوت بودن سلیقه‌های افراد در انتخاب مرورگر، ممکن است سایتی با مرورگرهای مختلفی مشاهده شود. این وظیفه طراح است که برنامه را طوری طراحی کند که اولاً با مرورگرهای مختلف مشکلی نداشته و در همه آنها تا حد ممکن بدون تغییر ظاهر شود و ثانياً انعطاف‌پذیری لازم را به هنگام نمایش در ابعاد مختلف نمایشگر داشته باشد.

**برنامه‌نویسی برای تلفن همراه**

امروزه یکی از پرطرفدارترین وسایل، تلفن همراه است که به دلیل این محبوبیت، برنامه‌نویسی و تولید نرم‌افزار برای آن نیز از اهمیت زیادی برخوردار است. انواع بسیار زیاد و متنوعی از گوشی‌های تلفن همراه در بازار وجود دارند که هر یک از خانواده بخصوصی هستند. این گوشی‌ها معمولا از سیستم عامل‌های مختلفی بهره می‌گیرند. برنامه‌نویسی که در زمینه تلفن‌های همراه فعالیت می‌کند باید در نظر داشته باشد که یک برنامه همواره روی همه گوشی‌ها اجرا نخواهد شد و این سازگاری مستقیما با نوع گوشی و سیستم عامل موجود در آن و پشتیبانی سیستم عامل از زبان برنامه شما در ارتباط است. بهتر است برنامه‌نویس، زبانی را انتخاب کند که با سطح وسیع‌تری از سیستم عامل‌ها سازگار بوده و از جانب گوشی‌های بیشتری پشتیبانی شود. البته این مساله مربوط به زمانی است که فردی بخواهد برای گستره بیشتری از مخاطبان برنامه بسازد، وگرنه بسیاری از افراد تنها برای یک نوع گوشی خاص با سیستم عاملی مشخص برنامه می‌نویسند.

مساله دیگری که در این زمینه وجود دارد ابعاد نمایشگر است که این بار در مقایسه با برنامه‌سازی تحت وب از تنوع بیشتری برخوردار است. در این حالت برنامه‌نویس چند راه‌حل خواهد داشت. یکی این که سرویس‌دهنده و کاربران همگی اعضای یک شبکه داخلی هستند.

زمانی که به همه جنبه‌های برنامه‌نویسی توجه کافی شود، در نهایت به برنامه‌ای خواهیم رسید که از همه نظر مخاطبان خاص خود را راضی نگه خواهد داشت و پاسخگوی سلیقه و نیاز آنها خواهد بود.

ارسال شود. کاهش می‌یابد.

**سرعت و کارایی**

سرعت پایین برنامه‌های جاوا در محیط اجرایشان ملاک خوبی برای کارایی آن نیست، زیرا در محیط وب کندی بیشتر به علت شبکه است و در گام نخست باید سربرار شبکه را از روی برنامه‌ها برداشت. وگرنه در برنامه‌های کاربردی پهنه‌سازی‌های بسیاری به وجود آمده است که موجب شده در آخرین آزمون کارایی، یک برنامه جاوا از ۰/۸ تا ۱/۳ همان برنامه در C++ سرعت داشته باشد که ۱/۳ مربوط به بخش واسط کاربری(GUI) و سرعت ۰/۸ آن مربوط به تخلیه حافظه (Collection Garbage) می‌شده است و تاکنون هیچ الگوریتمی نتوانسته است در سرعت از الگوریتم تخلیه حافظه جاوا پیشی بگیرد.

ضمنا ثابت شده است که تجربه برنامه‌نویسی که برنامه‌ای را می‌نویسد از انتخاب زبانی که برنامه با آن نوشته می‌شود تاثیر بیشتری در کارایی دارد و این بدان معناست که کارایی یک برنامه را برنامه‌نویس و روش‌های به کار رفته در برنامه وی مشخص می‌کند و نه زبانی که با آن برنامه می‌نویسد. حذف اشاره‌گرها در جاوا به دلیل مشکلاتی بوده است که آنها در طول تاریخ استفاده‌شان به وجود آورده‌اند. اگرچه این موارد در برنامه‌های سیستمی لازم به نظر می‌رسد اما در محیط‌های تحت وب که بستر اصلی جاوا است می‌توانند اثراتی به مراتب شدیدتر نسبت به آنچه در برنامه‌های سیستمی دارند داشته باشند و باعث می‌شود که توجه برنامه‌نویسان از مسائلی چون کارایی، قابلیت اطمینان و مقیاس‌پذیری برنامه به تنظیم اشاره‌گرها معطوف شود.

در زبانی مانند C++ مفهومی به نام وراثت چندگانه وجود دارد که باعث ایجاد مشکلاتی اساسی می‌شد که اکثر برنامه‌نویسان C++ از آن

ابعاد مختلف بازنویسی کند. این اولین راه‌حلی است که ممکن است به نظر برسد، اما دو راه دیگر وجود دارد. مستقیما از ابعاد نمایشگر، ظاهر خود را حفظ کرده و در برابر تغییرات ابعاد از نظر ظاهری از خود انعطاف‌پذیری نشان دهد. دیگر آن که پیش از اجرای بدنه اصلی برنامه، ابعاد نمایشگر تشخیص داده شود و بعد از آن به بخشی که مربوط به آن ابعاد است پرش شود. در واقع نگارش‌های مختلف برنامه در قالب یک برنامه ارائه می‌شوند و با توجه به نوع گوشی، نگارش مورد نظر فراخوانی می‌شود. در این روش ممکن است نسبت به روش اول اندکی افزایش حجم داشته باشیم، اما در عوض تصمیم‌گیری و انتخاب نگارش برنامه را از کاربر به برنامه تغییر داده‌ایم که این موجب انعطاف‌پذیری بیشتر برنامه خواهد شد.

**برنامه‌نویسی تحت شبکه**

در شبکه سرعت ارتباطات و اجرای برنامه‌ها از اینترنت بیشتر اما از سرعت اجرای برنامه‌های مستقل کاربردی قابل اجرا در یک سیستم کمتر است. طبیعی است که در طراحی این نوع برنامه‌ها باید به هر دو دسته خصوصیات برنامه‌های تحت وب و برنامه‌های مستقل توأما توجه داشت و بسته به نوع نیاز به آن ویژگی اهمیت داد.

با توجه به شکل طراحی شبکه و نوع آن، همچنین رسانه‌های انتقال و سرعت آنها باید برنامه‌ها طوری طراحی شوند که سرعت برنامه‌های کاربردی مستقل و انعطاف‌پذیری برنامه‌های تحت وب را داشته باشند. ضمن این‌که می‌توان برای برنامه‌نویسی تحت وب عمل کرد. یعنی فرض را بر این گذاشت که قرار است برای وب برنامه‌ای طراحی کنیم که کاربران با اتصال به سرویس‌دهنده از آن استفاده می‌کنند. با این تفاوت که سرویس‌دهنده و کاربران همگی اعضای یک شبکه داخلی هستند.

زمانی که به همه جنبه‌های برنامه‌نویسی توجه کافی شود، در نهایت به برنامه‌ای خواهیم رسید که از همه نظر مخاطبان خاص خود را راضی نگه خواهد داشت و پاسخگوی سلیقه و نیاز آنها خواهد بود.

دوری می‌کرده و می‌کنند. ولی قابلیت چندریختی(Polymerphism) یک کلاس از لحاظ شی، گرایبی بسیار مهم بوده و بنابراین توجهی برای وجود وراثت چندگانه را فراهم می‌کرد.

در جاوا با وارد شدن مفهومی به نام Interface ، دیگر نیازی به وجود وراثت چندگانه احساس نشد و این مفهوم از زبان جاوا حذف شد. درحال حاضر طراحان برنامه‌ها حتی به این نتیجه رسیده‌اند که وراثت معمولی هم باعث ایجاد مشکل بوده و تا آنجایی می‌که می‌شود باید از مفهومی به نام ترکیب (Composition) استفاده کرد.

**ضعف جاوا**

مهم‌ترین اشکالی که برنامه‌نویسان سایر زبان‌ها به جاوا می‌گیرند سرعت اجرایی بسیار پایین آن است. یک برنامه جاوا به صورت بایت‌کد(byte code) است که کد زبان ماشین مجازی است و باید در آن محیط اجرا شود. به همین دلیل سرعت اجرای پایینی در مقابل زبان‌هایی چون C++ دارد. به عنوان مثال ممکن است یک برنامه به زبان C++ به‌طور متوسط تا ۱۰ برابر سریع‌تر از برنامه مشابه جاوا اجرا شود. جاوا برخلاف شی،گرا بودن، در بخشی از قسمت‌ها برای ایجاد انعطاف بیشتر، برخی از اصول شی،گرای را نادیده گرفته است که می‌توان آن را نیز از نقاط ضعفش برشمرد.

همان طور که پیش از این نیز اشاره کردیم، زبان جاوا نسبت به C++ ساده‌تر است و به همین دلیل آموزش دادن و آموختن آن نیز راحت‌تر است که البته به دلیل به وجود آمدن این سادگی‌ها بسیاری از نقاط قوتی که در C++ وجود داشته‌اند در جاوا حذف شده‌اند که مهم‌ترین آن مفهوم اشاره‌گر(Pointer) است که به آن اشاره کردیم. ضمنا با توجه به این که جاوا زبانی مستقل از سیستم‌عامل است، استفاده از توابع API سیستم‌عامل در آن امکان‌پذیر نیست.

نوشتن برنامه‌های تحت وب در اختیار برنامه‌نویسان قرار می‌دهند. این خصوصیتی است که جاوا را از بسیاری از زبان‌های برنامه‌نویسی متمایز می‌سازد.

البته برخلاف جاوا اسکریپت که برای اجرا نیاز به هیچ نوع نرم‌افزار اضافه نداشت، برنامه‌های تحت وبی که با استفاده از اپلت‌های جاوا نوشته شده‌اند نیاز به ماشین مجازی جاوا نرم‌افزار(JVM) دارند و بدون آن قادر به اجرا نیستند. (اگر به خاطر داشته باشید پیش از این گفته بودیم که جاوا و جاوا اسکریپت دو زبان کاملا متفاوت عرضه شده‌از دو شرکت مختلف هستند و تنها از نظر نام و تا حدودی شکل ظاهری کدها شبیه به یکدیگرند)
اپلت‌ها شبیه فناوری اکتیوایکس(ActiveX) شرکت مایکروسافت هستند و برنامه‌نویس را قادر می‌سازند تا امکانات جدیدی به مرورگر کاربر اضافه کند. البته اپلت‌ها به‌دلیل آن‌که در محیطی تحت‌عنوان Sand box اجرا می‌شوند امنیت دارند اما اکتیوایکس چنین امنیتی ندارد.

همان‌طور که قبلا نیز اشاره کردیم سرویس‌دهنده وب از طریق دستورات HTML که در صفحات وب وجود دارد، تعامل بین سرویس‌دهنده و کاربر را کنترل می‌کند. این کار در سایت‌هایی که مشتریان زیادی دارد، سرویس‌دهنده وب را دچار مشکل می‌کند. بهتر است کنترل توسط مرورگر انجام شود و سرویس‌دهنده نقش پشتیبان را داشته باشد. ضمنا رایانه کاربران معمولا بیکار است و منتظر پاسخگویی سرویس‌دهنده می‌ماند. یک روش حل این مساله آن است که بعضی از تعامل‌های سرویس‌دهنده در محیط مرورگر انجام شود. به جای انتقال داده‌های HTML به سرویس‌دهنده، سرویس‌دهنده می‌تواند یک برنامه کوچک را به مرورگر بفرستد و برنامه کاربردی را در ماشین مجازی اجرا کند. به این ترتیب حجم اطلاعاتی که باید از سرویس‌دهنده به مرورگر