

How to Become A Successful Java Developer



ویروس کرونا باعث لغو بسیاری از کلاس‌های آموزشی شده و تقریباً تمامی موسسات کلاس‌های خود را به صورت مجازی برگزار می‌کنند. برگزاری کلاس‌ها به شیوه آنلاین مزایا و معایب خاص خود را دارد. اگر در دوره‌های آنلاین شرکت کنید و مربی خوبی داشته باشید، تقریباً خیالتان از بابت یادگیری مهارت‌ها آسوده است، البته تنها مشکلی که در این زمینه دارید این است که مجبور هستید پهنای باند اینترنت خود را برای این کار اختصاص دهید و ملزومات مورد نیاز را تهیه کنید. اگر خودتان به فکر یادگیری هستید، اوضاع فرق می‌کند و انعطاف‌پذیری بیشتری در ارتباط با یادگیری در اختیارتان قرار می‌گیرد.

اگر به فکر یادگیری جاوا به شیوه شخصی هستید باید روی مباحث زیر متمرکز شده و مدت زمان قابل توجهی را به این مفاهیم اختصاص دهید.

1. به فکر یادگیری اصول زیربنایی و مفاهیم برنامه‌نویسی شی‌گرایی باشید

برای یک توسعه‌دهنده جاوا مهم است که درک درستی از برنامه‌نویسی شی‌گرایی داشته باشد. بدون داشتن پایه و اساس قوی در برنامه‌نویسی شی‌گرایی این امکان وجود ندارد تا بتوانید زبان جاوا را به درستی یاد بگیرید، زیرا بخش عمده‌ای از مفاهیم جاوا بر مبنای مفاهیم شی‌گرایی هستند. اگر تجربه برنامه‌نویسی شی‌گرایی دارید، بازهم لازم است برای تسلط بر جاوا اطلاعات بیشتری در ارتباط با اصول شی‌گرایی در جاوا به دست آورید. دقت کنید که مطالعه صرف تعاریف اصلی شی‌گرایی کمک زیادی به شما نمی‌کند، بلکه باید بدانید که چگونه از این اصول در طراحی راه‌حل‌ها به شیوه شی‌گرایی استفاده کنید. به همین دلیل مهم است که اطلاعات دقیقی در مورد پیاده‌سازی مفاهیمی همچون اشیاء، وراثت، چند ریختی، الگوهای طراحی و... به دست آورید.

2. باید دانش کافی در مورد واسط‌های برنامه‌نویسی اصلی جاوا به دست آورید

مهم نیست که به لحاظ دانش نظری چقدر قوی هستید، زیرا بدون تسلط بر ساختارهای زبان و واسط‌های اصلی برنامه‌نویسی قادر به انجام کار خاصی نیستید. در ارتباط با جاوا باید تجربه کاربردی زیادی در ارتباط با API‌های اصلی همچون `java.lang`، استثناءها، مجموعه‌ها، `Generics`، ریسمان‌ها، `I/O`، `JDBC` و... به دست آورید. زمانی که صحبت از ساخت برنامه‌های تحت وب به میان می‌آید مهم نیست از چه چارچوبی استفاده می‌کنید. مهم این است که مفاهیم مربوط به سرویس‌ها، `JSB` را درک کنید. این یک اصل ضروری است.

3. کدنویسی را متوقف نکنید

زمانی که در مورد مفاهیم و مباحث تئوری صحبت می‌کنیم همه چیز ساده به نظر می‌رسد. ما به لحاظ تئوری

می‌توانیم به سادگی یک راه‌حل را ارائه کنیم، اما نمی‌توانیم تا زمانی که به شکل عملی کاری را شروع نکرده‌ایم عمق یک موضوع را به درستی درک کنیم. مادامی که برنامه‌نویسی به شکل عملی آغاز نشود با محدودیت‌های آن زبان آشنا نخواهید شد و مهم‌تر آن‌که شناخت درستی از بهترین روش‌ها و الگوهای طراحی آن زبان به دست نمی‌آورد. بنابراین نباید کدنویسی را متوقف کنید.

4. در انجمن‌های کدنویسی مشترک شوید

شما تنها فردی نیستید که تصمیم گرفته‌اید جاوا را یاد بگیرید، افراد زیادی هستند که روی فناوری‌هایی کار می‌کنند که ما روی آن‌ها متمرکز شده‌ایم. در حالی که کار کردن با یک چهارچوب ممکن است یک چالش واقعی نباشد، اما زمانی که تصمیمی دارید روی یک پروژه واقعی کار کنید با مسائل عجیبی روبرو می‌شوید که در مستندات رسمی هیچ راه‌حلی برای آن‌ها وجود ندارد. در زمان متمرکز شدن روی یک فناوری جدید، بهترین و اولین کاری که باید انجام دهید عضویت در انجمن‌ها مرتبط با آن فناوری است. همواره این احتمال وجود دارد که در زمان مواجه شدن با یک مشکل، شخصی دیگری در این کره خاکی با آن مشکل روبرو شده و برای آن راه‌حلی پیدا کرده باشد. پاسخ‌هایی که پرزش‌های افراد در انجمن‌ها داده شده را مطالعه کنید، زیرا این پرسش‌ها و پاسخ‌ها به مسائل جالب و مهمی اشاره دارند.

5. وبلاگ‌ها و پاسخ‌ها را دنبال کنید

همان‌گونه که اشاره کردم، شما تنها نیستید. هزاران نفر در سراسر جهان از فناوری‌هایی مشابه با آن چیزی که شما روی آن متمرکز شده‌اید استفاده می‌کنند و نقطه نظارت خود را درون وبلاگ‌های شخصی درج می‌کنند. با توجه به این‌که اطلاعات درون وبلاگ‌ها بر مبنای تجربیات توسعه‌دهندگان تنظیم شده، مطالعه اطلاعات مندرج در این وبلاگ‌ها بینش شما در ارتباط با فناوری‌های مورد استفاده را افزایش می‌دهند. برخی از توسعه‌دهندگان ممکن است یک فناوری را ارزشمند توصیف کنند، در حالی که برخی دیگر آن‌را کم ارزش بدانند. بر مبنای این دیدگاه قادر هستید خوب و بد فناوری‌های مختلف را بشناسید. راه دیگر به دست آوردن اطلاعات ارزشمند از طریق مطالعه نظرانی است که در انتهای پست‌های وبلاگی قرار گرفته‌اند.

6. کد منبع مربوط به چارچوب‌های منبع باز را مطالعه کنید

یک توسعه‌دهنده خوب باید نحوه استفاده از چهارچوب‌ها را یاد بگیرد. اگر تمایل دارید یک توسعه‌دهنده برجسته شوید باید کد منبع چارچوب‌های موفق را مطالعه کنید تا نحوه کار و مکانیزم داخلی چارچوب‌ها را به خوبی یاد بگیرید. بررسی کدهای منبع در نحوه استفاده از توابع درون چارچوب‌ها کمک فراوانی می‌کنند.

7. روندهای نوظهور فناوری را بررسی کنید

از منظر روند توسعه نرم‌افزارهای منبع باز، آن‌ها به‌طور چشم‌گیری در حال رشد هستند. زمانی که روی یادگیری یک فناوری یا چارچوب جدیدی متمرکز می‌شوید، مهم است که بدانید همواره این احتمال وجود دارد که برخی از ویژگی‌های درون یک چارچوب منسوخ شوند و ویژگی‌های جدیدی جایگزین آن شوند. در برخی موارد ممکن است مشکلی که قرار است با کدهای زیادی توسط یک چارچوب حل شود در نسخه جدید تنها با یک خط حل شوند. به همین دلیل لازم است نسخه‌های جدید یک فناوری را بررسی کنید.

8. کدها و ابزارهای دم دستی را در مکان در دسترس قرار دهید

با گذشت زمان، ممکن است به کپی کردن کدهایی که قبلاً نوشته‌اید نیاز پیدا کنید. به‌طور مثال، اطلاعات پیکربندی شبیه به `log4.properties`، پیکربندی JDBC و ابزارهایی شبیه به `DBUtils`، `ReflectionUtils`، `StringUtils` و... را در دست نگه دارید. به‌طور مثال، ممکن است در برخی موارد مجبور شوید فهرستی از مقادیر درون یک خاصیت را از درون یک مجموعه از اشیا واکشی کنید، در چنین شرایطی از `ReflectionUtil` برای این منظور استفاده می‌کنید. رویکردی که تنها چند دقیقه از وقت شما را می‌گیرد، البته به شرطی که نحوه استفاده از این شی را به خوبی یاد گرفته باشید.

9. متدولوژی‌های مختلف توسعه را بررسی کنید

بهتر است در مورد متدولوژی‌های مختلف شبیه به اسکرام، آبشاری، چابک و XP اطلاعات لازم را به دست آورید. این

