

به نام خدا

مقدمه، آموزش اول

شاید اگر سال ۲۰۱۳ را سال برنامه نویسان بنامیم خیلی اغراق نکرده باشیم چرا که در این سال برنامه نویسی در رأس هرم برترین مشاغل سال و فرصت های شغلی قرار گرفته است. در حقیقت این مهم نیست که ما چه زبانی را برای برنامه نویسی انتخاب می کنیم بلکه نکته ای که می بایست مد نظر قرار داده شود این است که در این مسیر لذت بخش و درآمد زاد گام نهاده و جامعه ای که در آن زندگی می کنیم را از یک سو و زندگی شخصی خود را از سوی دیگر متحول سازیم. اینکه چرا زبان برنامه نویسی جاوا را مد نظر قرار داده ایم سوالی است که با این پاسخ روشن خواهد شد:

بدون اغراق زبان برنامه نویسی جاوا به منزله یکی از معروف ترین زبان های برنامه نویسی در دنیا است چرا که از این زبان برنامه نویسی علاوه بر ساخت برنامه های کامپیوتری، در ساخت اپلیکیشن ها و بازی های اندروید نصب شده روی تلفن های همراه و تبلت ها و همچنین برنامه نویسی برای وب سایت های تعاملی مثل وب سایت بانک ها و موسسات بزرگ نیز استفاده می شود. در حقیقت یکی از علل میزان محبوبیت این زبان برنامه نویسی در میان میلیون ها برنامه نویس حرفه ای در سرتاسر جهان این است که این زبان برنامه نویسی غیرممکن ها را ممکن ساخته است. هیچ جای شگفتی نیست که آشنایی با زبان برنامه نویسی پلتفرم محبوبی همچون اندروید که بیش از ۷۰ درصد سهم بازار جهانی را در میان تلفن های همراه به خود اختصاص داده است می تواند امید بخش آینده مالی خوبی نیز برای برنامه نویسان این حوزه باشد. زبان برنامه نویسی جاوا زبان ساده ای است مخصوصاً برای کسانی که از پیشینه برنامه

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

نویسی با C و C++ برخوردار باشند. در زبان جاوا شیء گرایی مورد استفاده قرار می گیرد به این شکل که هر چیزی در زبان جاوا یک شیء است و همین مسئله برنامه نویسی در محیط جاوا را بسیار ساده می سازد. یکی از ویژگی های منحصر به فرد جاوا مدیریت حافظه است که حافظه تخصیص داده شده را مدیریت کرده و به موقع آزاد می کند. جاوا این امکان را به ما می دهد تا در آن واحد چندین برنامه را به طور هم زمان اجرا کنیم. از دیگر مزایای این زبان برنامه نویسی می توان امنیت بالای برنامه هایی اشاره کرد که با جاوا نوشته می شوند چرا که جاوا با ایجاد یک Firewall مابین کامپیوتر شما و شبکه، از حمله ویروس ها به اطلاعات شخصی شما جلوگیری به عمل خواهد آمد. برنامه های نوشته شده توسط جاوا قابلیت حمل دارند و روی هر سیستمی که جاوا روی آن نصب باشد قابل اجرا است.

در واقع در این سری آموزش های زبان برنامه نویسی جاوا حداکثر تلاش صورت گرفته است تا آموزش ها برای گروهی از مخاطبین نگارش شوند که تاکنون حتی یک خط کد هم در زندگی خود ننوشته اند. به عبارت دیگر این امید را به خوانندگان عزیز می توان داد که چنانچه یک رایانه شخصی، کمی زمان آزاد اما در عین حال انگیزه بسیار زیادی داشته باشند قادر گردند پس از خواندن کلیه آموزش ها برای مدیریت زندگی شخصی خود، شرکت خود و یا دوستان و یا برای شرکتی که در آن کار می کنند برنامه ای مفید بنویسند.

شاید بارها و بارها به فکر این افتاده باشید که یک زبان برنامه نویسی را فرا بگیرید اما به دلیل اینکه شنیده اید که برنامه نویس شدن نیازمند برخورداری از یک ذهن تحلیلی، داشتن یک مدرک مرتبط از دانشگاه و داشتن جدید ترین رایانه شخصی چه از لحاظ

نرم افزاری و چه از لحاظ سخت افزاری است از این فکر صرف نظر کرده اید. خبر خوشحال کننده برای آن دسته از علاقمندان که تاکنون چنین طرز فکری داشته اند این است که عدم برخورداری از هیچ کدام از موارد فوق الذکر مانع برنامه نویسی شدن شما نخواهد شد. شاهد مثال این ادعا هم این است که نگارنده این سری از آموزش ها هیچ کدام از موارد فوق به جزء تا حدودی مورد اول را نداشته است. نکته جالب توجه آن جا است که پس از فراگیری اصول برنامه نویسی شما از یک ذهن تحلیلگر برخوردار خواهید شد که نه تنها در حوزه برنامه نویسی شما سودمند خواهد بود، بلکه در زندگی روزمره شخصی شما نیز راه کارهای بیشماری را در ارتباط با مسائل مختلف پیش روی شما خواهد گذاشت. علاوه بر این مورد شما به یک سرمایه علمی دست خواهید یافت که هرگز با گذراندن واحدهای آکادمیک دانشگاه به آن دست نخواهید یافت چرا که آنچه در این دوره فرا خواهید گرفت برنامه نویسی عملی و کاربردی است و تئوری تا جایی مورد استفاده قرار می گیرد که منجر به درک بهتر اصول برنامه نویسی جاوا گردد. و در نهایت سودی که از برنامه نویسی عاید شما می شود به طور حتم به گونه است که شما را قادر خواهد ساخت هر کامپیوتر و یا لپ تاپی که تاکنون آرزوی آن را داشته اید را به سهولت خریداری نمایید.

مقدمه بر پیشینه زبان جاوا

زبان برنامه نویسی جاوا در اوایل دهه ۱۹۹۰ توسط جیمز گاسلینگ و همکارانش در شرکت سان مایکرو سیستمز پایه ریزی شد که در ابتدا Oak به معنی بلوط نام داشت و علت این نام گذاری این بود که جیمز گاسلینگ در حین نوشتن این زبان برنامه نویسی جدید از پنجره اتاقش به یک درخت بلوط نگاه می کرد. اما پس از مدتی مدیران شرکت سان مایکرو سیستمز تصمیم گرفتند نام بهتری برای این زبان برنامه نویسی جدید انتخاب نمایند از این رو طراحان این زبان به کافی شاپی در نزدیکی شرکت رفتند تا نام جدید را

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

انتخاب نمایند. پس نیمی از روز بحث و تبادل نظر نام Java که مخفف نام های James Gosling و Arthur Van Hoff و همچنین Andy Bechtolsheim بود انتخاب شد و از آنجا که مراسم نام گذاری در یک کافی شاپ صورت گرفته بود نماد این زبان یک فنجان قهوه داغ در نظر گرفته شد. شرکت سان مایکرو سیستمز بعد ها توسط شرکت بزرگ اوراکل خریداری شد. از همان ابتدا شعار طراحان این زبان نامه نویسی این بود که یک بار بنویس، همه جا اجراش کن. در حقیقت آقای گاسلینگ از اصول برنامه نویسی در C++ خیلی احساس رضایت نمی کرد و یک سری نارسایی های این زبان برنامه نویسی موجب شد که وی زبان جاوا را بر مبنای زبان C++ طراحی کند به نحوی که برنامه های مد نظر ایشان را به نحو بهتری اجرا کند. این زبان برنامه نویسی موفق که در سال ۱۹۹۵ به طور رسمی به بازار معرفی شد توانست چنان محبوبیتی در جهان پیدا کند که در حال حاضر در بیش از ۳ میلیارد ابزار رایانه ای و تلفنی مورد استفاده قرار گرفته و تاکنون بیش از ۱۰۰۰ جلد کتاب پیرامون آن به رشته تحریر درآمده است. این زبان داری ساختار یا Syntax مشابه زبان های C و C++ می باشد اما از سادگی بیشتری برخوردار است. اگر چه که زبان برنامه نویسی جاوا نسبت به زبان برنامه نویسی Visual Basic از دشواری بیشتری برخوردار است اما این زبان از مزایای بسیار زیاد دیگری برخوردار می باشد که آن را قابل رقابت با دیگر زبان های برنامه نویسی ساخت است. از سوی دیگر زبان برنامه نویسی جاوا شیء گرا است یا اصطلاحاً یک زبان Object Oriented Language است. در زبان برنامه نویسی جاوا بر خلاف زبانی همچون FORTRAN که مجموعه ای از دستورات محض است، ما با Object ها یا اشیائی سرو کار داریم که حاوی داده هستند. در واقع در زبان جاوا هم ما یکسری دستورات می نویسیم اما تفاوت آن با زبانی همچون زبان برنامه نویسی فوق الذکر در این است که در جاوا ما ابتدا اطلاعات را دسته بندی کرده سپس بر اساس آن

ها دستوراتی را ترتیب می دهیم. در زبان های برنامه نویسی شی گرا ما با Object ها یا اشیاء و همچنین Class ها یا کلاس ها سرو کار داریم. در آموزش های آتی به طور مفصل تری با مفاهیم شی گرایی آشنا خواهیم شد اما در اینجا برای روشن تر شدن مطلب به ذکر دو مثال ملموس از دنیای واقعی اکتفا می کنیم.

فرض کنیم که قرار است ما یک آپارتمان ۱۰ طبقه بسازیم. هر ۱۰ طبقه از آپارتمان ما دارای یکسری ویژگی های خاص و منحصر به فرد است مثل اینکه رنگ دیوار های طبقه اول سفید است و درهای آن کرم رنگ هستند اما این در حالی است که رنگ دیوار های طبقه هفتم صورتی است و درهای آن سفید است و ... در حقیقت هر یک از طبقه های آپارتمان ما همچون یک Object یا شی در زبان برنامه نویسی جاوا است. اگرچه که هر یک از طبقات آپارتمان ما دارای یکسری ویژگی های خاص است، اما این در حالی است که کلیه طبقات آپارتمان از یکسری ویژگی های ثابت همچون اتاق خواب، سرویس، آشپزخانه و پنجره برخوردارند. در زبان برنامه نویسی جاوا یک Class دقیقاً چنین نقشی را ایفا می کند. به عبارت دیگر در یک برنامه تحت جاوا، Class به منزله یک نقشه خانه است که هر یک از طبقات آپارتمان از روی آن ساخته می شود اما هر یک از طبقات علیرغم داشتن یک پلان ثابت دارای یکسری ویژگی های منحصر به فرد خود است که اصطلاحاً به هر یک از آن طبقات یک Object گفته می شود. حال در برنامه نویسی، برنامه نویس یک Class با یکسری خصوصیات تعریف کرده سپس از روی آن Class تعریف شده تعدادی Object می سازد و این Object ها هر کدام دارای زیربنایی یکسان اما نمایی متفاوت می باشند. در واقع این همان ویژگی است که زبان برنامه نویسی جاوا را بسیار منحصر به فرد کرده است. آنچه زبان برنامه نویس جاوا را بسیار زیبا می سازد این ویژگی است که در ذیل در قالب مثال آپارتمان ذکر می شود. در واقع ما یک نقشه کلی یا Class برای آپارتمان طراحی می کنیم. حال تصمیم می

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

گیریم که تعدادی از طبقات آپارتمان ما سه خوابه بوده و تعدادی از آنها چهار خوابه ساخته شوند. در حقیقت ما از روی Class اصلی دو Class دیگر تحت عناوین سه خوابه و چهار خوابه طراحی می کنیم که ویژگی های Class اصلی را به ارث می برند. در برنامه نویسی جاوا به این رویداد Inheritance یا به ارث بردن می گویند. اصطلاحاً Class اولیه Superclass یا Class اصلی است و کلاس های سه خوابه و چهار خوابه Subclass یا Class زیرمجموعه می باشند. به عبارت دیگر می توان گفت که Class اصلی Class والد بوده و Class های زیرمجموعه Class فرزند می باشند. نکته ای که در اینجا می بایست حتماً مد نظر قرار دهیم این است که Class خود به تنهایی منجر به ایجاد چیزی نخواهد شد چرا که Class یک مفهوم انتزاعی است و چنانچه بخواهیم نمود عینی یک Class را ببینیم به یک Object نیاز داریم. در حقیقت پیش از طرح ریزی زبان های برنامه نویسی شئی گرا، برنامه نویسان چنانچه مجبور بودند تغییری در بخش از برنامه خود ایجاد کنند می بایست کل برنامه را از ابتدا بنویسند اما پس از ظهور زبان های برنامه نویسی شئی گرا این مشکل به کلی حل شد چرا که با اعمال تغییر روی یک Class مد نظر، کلیه Object های برگرفته از آن Class تغییر خواهند یافت.

در مثال دوم فرض کنیم که قرار است برای یک بانک یک نرم افزار حفظ اطلاعات حساب های مشتریان بنویسیم. ساختار چنین برنامه ای به شکل زیر خواهد بود:

Account		
Name	Id	Balance
CustomerOne	2234	1000000\$
CustomerTwo	2235	2000000\$

در مثال فوق ما یک Class داریم تحت عنوان Account و این کلاس ما دارای سه متغیر با نام های Name, Id, Balance به معنی به ترتیب نام، شناسه و موجودی

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

می باشد. حال از روی این Class می توانیم دو Object ایجاد کنیم که Object اول دارای خصوصیات

Name: CustomerOne, Id: 2234, Balance: 100000

و Object دوم دارای خصوصیات

Name: CustomerTwo, Id: 2235, balance: 2000000 می باشد.

در واقع هر دو Object ما نمود عینی کلاس Account می باشند. امیدوارم با ذکر این دو مثال مفهوم برنامه نویسی شئی گرایی مشخص شده باشد و در صورتیکه باز هم بخشی از مفاهیم آن طور که باید و شاید ملموس نیستند هیچ جای نگرانی نیست چرا که در آموزش های آتی در طول دوره این مفاهیم به خوبی منتقل خواهند شد.

هر برنامه کامپیوتری یا به عبارتی هر نرم افزار مجموعه ای از دستوراتی است که به ترتیب خاصی به کامپیوتر داده می شوند و یکی پس از دیگری اجرا می شوند. به طور مثال یک ویروس که برنامه مخرب کامپیوتری است که دستورات آن بر مبنای تخریب و یا از بین بردن بخشی از اطلاعات ذخیره شده روی کامپیوتر نوشته شده اند. برای روشن تر شدن مطلب، نرم افزار را به یک مثال از زندگی روزمره مان تشبیه می کنیم. فرض را بر این بگذاریم که خودروی شخصی شما خراب می شود و شما مجبور می شوید که آن را نزد مکانیکی ببرید. مکانیک پس از عیب یابی خودروی شما از شاگرد خود می خواهد که موارد ذیل را به ترتیب اجرا کند:

اول ماشین را خاموش کند. دوم کاپوت ماشین را بالا بزند. سوم روکش انژکتور را بردارد. چهارم پس از باز کردن انژکتور سوزن سوم آن را تعویض کند. پنجم انژکتور را مجدداً سوار کند. ششم روکش آن را نصب کند. و در نهایت کاپوت ماشین را بسته و استارت بزند.

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

در مثال فوق مکانیک به منزله برنامه نویسی است و شاگرد هم به منزله اجرا کننده و یا همان کامپیوتر. مفهوم تعمیر شدن ماشین را نیز می توان به عملیاتی تشبیه کرد که یک نرم افزار انجام می دهند. هر یک از دستوراتی را که مکانیک به شاگردش می دهد همان دستوراتی است که ما به عنوان برنامه نویسی به کامپیوتر می دهیم تا کاری را انجام دهد. حال اگر این دستورات به نحوی باشند که کار مثبتی صورت گیرد ما یک نرم افزار تولید کرده ایم و چنانچه این دستورات موجب ایجاد خللی در سیستم شوند و به جایی صدمه بزنند ما یک بدافزار یا ویروس طراحی کرده ایم. به عبارت دیگر شاگرد مکانیک به جای تعوض سوزن انژکتور می توانست نه تنها سوزن را تعویض نکند بلکه از عمد به مابقی سوزن ها هم صدمه بزند.

در برنامه نویسی اصطلاحاً به هر دستوری که می خواهیم اجرا شود یک Statement گفته می شود. حال برای نوشتن تعدادی Statement ما نیاز به یک ویرایشگر متن داریم تا بتوانیم دستورات مد نظر را در آن وارد سازیم.

انتخاب ابزاری برای برنامه نویسی جاوا

محیط های برنامه نویسی یا IDE های زیادی همچون JDK، Eclipse، IntelliJ IDEA و NetBeans هستند که به منظور نوشتن کد های جاوا و دیگر زبان های برنامه نویسی می توان از آن ها استفاده کرد. در این سری از آموزش ها از محیط برنامه نویسی قدرتمند Eclipse استفاده می کنیم که علاقمندان می توانند به صورت رایگان آن از سایت <http://eclipse.org/downloads> بسته به نوع سیستم عامل خود دانلود کنند. یکی از دلایلی که شروع برنامه نویسی جاوا را با محیط برنامه نویسی اکلپس آغاز خواهیم کرد این است که چنانچه بخواهیم در آینده به برنامه نویسی اندروید پردازیم با هیچ گونه مشکلی مواجه نخواهیم شد زیرا با توجه به اینکه این نرم افزار

رایگان است شرکت گوگل هم شرایط نصب SDK اندروید را روی این نرم افزار فراهم نموده و ما برای طراحی اپلیکیشن های اندروید با هیچ گونه مشکلی مواجه نخواهیم شد. در واقع برنامه Eclipse نیازی به نصب نداشته و صرفاً پس از خارج کردن فایل دانلود شده از حالت زیپ با کلیک کردن بر روی آیکان eclipse.exe می توانید آن را اجرا نمایید. به منظور آماده سازی محیط برنامه نویسی جاوا شما حتماً به JRE و JDK نیاز دارید که می توانید آخرین نسخه آن ها را از لینک <http://www.oracle.com/technetwork/java/javase/downloads/index.html> دریافت نمایید. (لازم به ذکر است چنانچه شما نسخه JDK نصب نمایید دیگر نیازی به نصب JRE نخواهید داشت چرا که JDK در برگیرنده JRE نیز می باشد).

پس از اجرای برنامه اکلیپس از شما درخواست می شود تا مسیری را مشخص نمایید تا پروژه های جاوا که طراحی می کنید را در آن ذخیره سازد. این مسیر به دلخواه شما می تواند هر جایی رو کامپیوتر شما باشد. پس از مشخص کردن مسیر گزینه OK را انتخاب نمایید. از آنجا که این اولین باری است که شما نرم افزار را اجرا می کنید، یک صفحه خوش آمد گویی یا Welcome Screen نمایان خواهد شد. پس از نگه داشتن نشانگر موس خود روی تک تک گزینه ها، گزینه مرتبط با Workbench یا صفحه اصلی نرم افزار را پیدا نموده و روی آن کلیک نمایید. حال شما در محیط اصلی نرم افزار قرار دارید. از آنجا که ممکن است شما بیش از یک نسخه از نرم افزار جاوا را روی سیستم عامل خود نصب داشته باشید، نیاز است که در این مرحله برای اکلیپس مشخص کنید که از کدام نسخه جاوا برای برنامه نویسی استفاده کند (در حال حاضر آخرین نسخه جاوا ۷ می باشد). در منوی اصلی اکلیپس، با انتخاب گزینه Windows زیرشاخه Preferences را انتخاب نمایید. در قسمت چپ پنجره منوی Java را انتخاب

نمایید. با کلیک کردن روی مثلث مشکی رنگ کنار واژه Java قادر خواهید بود تا زیرشاخه های آن را نیز مشاهده کنید. از میان زیرشاخه ها گزینه **Installed JREs** را انتخاب کنید. پس از انتخاب این زیرشاخه در منوی سمت راست، پنجره ای را مشاهده خواهید کرد که کلیه نسخه های **JRE** جاوا که بر روی سیستم شما نصب است را لیست کرده است. به طور مثال چنانچه نسخه **JRE 7** بر روی سیستم عامل شما نصب باشد، شما آخرین نسخه جاوا که توسط شرکت اوراکل منتشر شده را بر روی سیستم خود داشته و قادر خواهید بود از کلیه قابلیت های جاوا در طراحی نرم افزار، اپلیکیشن اندروید و همچنین طراحی بازی استفاده نمایید.

حال چنانچه نسخه **JRE** مد نظر شما در لیست موجود نباشد می توانید روی دکمه **Add** کلیک نموده و آن را به لیست فوق اضافه نمایید. در پنجره ای که باز خواهد شد روی گزینه **Standard VM** کلیک نمایید و سپس دکمه **Next** را کلیک نمایید. حال در بخش **JER home** مکانی که در آن جاوا را روی سیستم عامل خود نصب نموده اید را مشخص نمایید. معمولاً نسخه های **JRE** که روی ویندوز نصب می شوند در مسیر دایرکتوری **C:\Program Files\Java\jre7** قرار می گیرند. در صورتیکه اکلیپس به صورت خود کار بخش **JRE name** را پر نکرد می توانید این کار را خود با نامی مرتبط با **JRE** مد نظر انجام دهید. حال می توانید گزینه **Finish** را انتخاب نمایید تا تغییرات شما اعمال شود و به صفحه ای بازگردید که در آن نسخه های نصب شده **JRE** شما قابل مشاهده هستند. نسخه نصب شده جدید خود را پیدا نموده و آن را انتخاب نمایید. در این مرحله کار شما به پایان رسیده است. حال در همان منوی اصلی جاوا که از آن **Installed JREs** را انتخاب نمودید، زیرشاخه **Compiler** را پیدا نموده و به هیچ وجه آن را باز ننمایید بلکه فقط با کلیک کردن موس روی آن، آن را انتخاب نمایید تا رنگ آن آبی شود. حال در منوی سمت راست مرتبط با زیرشاخه

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

Complier از بخش Complier compliance level لیستی از نسخه های نصب شده JRE بر روی سیستم عامل را خواهید دید. به طور مثال برای جاوای نسخه ۷ مورد ۱.۷ را از لیست انتخاب نمایید. حال می توانید دکمه OK صفحه Preferences را بزنید و به صفحه اصلی نرم افزار اکلیپس باز گردید. حال کلیه ابزارهای مورد نیاز برای نوشتن اولین برنامه جاوای خود را به صورت رایگان در اختیار دارید.

پیش نیاز ها

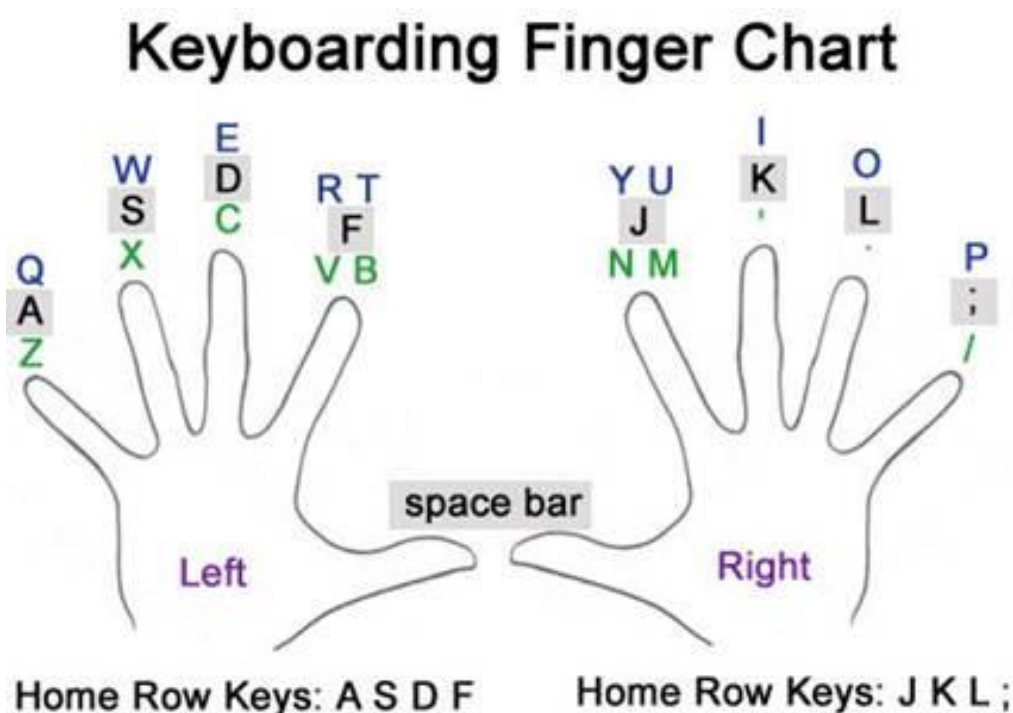
حال که ما تصمیم گرفته ایم به دنیای بی انتها و لذت بخش برنامه نویسی قدم بگذاریم، لاجرم نیاز داریم تا در کنار اصول و مفاهیم برنامه نویسی به علوم و مهارت های جانبی که به حرفه ما کمک می کنند نیز نیم نگاهی داشته باشیم. در حقیقت در کنار فراگیری زبان برنامه نویسی جاوا، ما می بایست تفکری انتقادی یا اصطلاحاً Critical Thinking و تفکر منطقی یا Logical Thinking داشته باشیم، با اصول و قواعد ریاضیات مثل الگوریتم آشنایی داشته باشیم، معانی رنگ ها و نحوه به کار گیری رنگ در اپلیکیشن را بدانیم و غیره. شاید بیان چنین مسئله ای تا حدودی نا امید کننده باشد که چه طور ممکن است فردی کلیه موارد فوق را به علاوه بسیاری موارد دیگر به طور هم زمان فرا گیرد. در این آموزش بنده قصد ندارم تا پیرامون مطالب فوق الذکر صحبت کنم بلکه قصد دارم به دو مهارتی اشاره کنم که شاید برای خیلی از دوستان مهم به نظر نرسد اما در دراز مدت می تواند حرفه شما را به عنوان یک برنامه نویس تحت الشعاع قرار دهد و آن دو مهارت عبارتند از فراگیری تایپ اصولی و زبان انگلیسی.

در واقع همانطور که یک IDE به منزله یک ابزار برنامه نویسی است، نحوه صحیح تایپ نیز می تواند کمک شایانی در حرفه ای شدن ما به عنوان یک برنامه نویس کند. از دید

بنده کسب مهارت تایپ صحیح (ده انگشتی) برای حرفه برنامه نویسی لازم و ضروری است. به عبارت دیگر برنامه نویسی صرفاً در نوشتن کد خلاصه نمی شود بلکه ما به عنوان یک برنامه نویس گاهی نیاز داریم تا پیرامون اپلیکیشن خود توضیحاتی بنویسیم و یا چنانچه عضو یک تیم برنامه نویسی باشیم نیاز داریم تا با مابقی برنامه نویسان در تعامل بوده و مکاتبه نماییم و یا از طریق Comment توضیحاتی را درون کد خود بنویسیم. در حقیقت چنانچه ما به تایپ سریع و صحیح تسلط داشته باشیم مابین فکر کردن و نوشتن کد زمان اندکی را صرف خواهیم کرد. شاید با ذکر این دو مثال اهمیت فراگیری تایپ سریع را بهتر متوجه شویم. خیلی اوقات اتفاق می افتد که ما با هدف خاصی به یکی از اتاق های منزل خود می رویم اما به محض اینکه به آن اتاق مد نظر می رسیم فراموش می کنیم که به چه منظور به آن اتاق رفته ایم (البته این مسئله در مورد خیلی از دوستان صدق نخواهد کرد). یا اینکه در بحثی شرکت کرده ایم و در همین حین که یکی از دوستان ما در حال صحبت کردن است نکته ای به ذهن ما می رسد و صبر می کنیم تا صحبت های دوستان به اتمام برسد اما همین که دوستان سکوت کرد ما نکته ای را که می خواستیم بیان کنیم را فراموش کرده ایم. مابین زمانی که ما تصمیم می گیریم تا به آن اتاق برویم و زمانی که به آن اتاق می رسیم ممکن است چندین ثانیه طول بکشد و فراموشی هم می تواند در همین فاصله زمانی اتفاق افتد. این قضیه پیرامون اظهار نظر کردن در مثال دومی هم که زدیم نیز صادق است. دقیقاً چنین اتفاقی هم در برنامه نویسی برای ما ممکن است پیش آید (توجه بفرمایید عرض کردم ممکن است). در واقع زمان مابین ایده ای که در برنامه نویسی به ذهن ما خطور می کند تا تبدیل ایده به کد می تواند منجر به فراموشی گردد. حال اگر بتوانیم این زمان را به حداقل برسانیم می توانیم انتظار داشته باشیم فراموشی ما هم به حداقل برسد. در واقع هدف ما در این مقاله این نیست که برنامه نویسان را مجبور کنیم که به کلاس تایپ حرفه ای بروند بلکه حداقل انتظاری که

می رود این است که تایپ صحیح را بیاموزند. تجربه نشان داده است زمانیکه ما با استفاده از ده انگشت دست خود اقدام به تایپ کردن می نمایم، نه تنها سرعت تایپ ما افزایش می یابد بلکه کمتر هم از اینکار خسته خواهیم شد چرا که میزان باری که روی دو انگشت وارد می آید (در تایپ دو انگشتی)، حال در تایپ ده انگشتی مابین ده انگشت تقسیم خواهد شد و بالتبع میزان خستگی ما نیز کمتر خواهد شد.

به منظور فراگیری نحوه صحیح تایپ راه کارهای متفاوتی وجود دارد من جمله نرم افزارهایی که برای این منظور طراحی شده اند و حتی یکسری کلاس های آموزشی نیز وجود دارند. اما راه کاری که بنده برای فراگیری تایپ ده انگشتی استفاده کردم عکس زیر بود:



در حقیقت ما با مد نظر قرار دادن این نکته که دو انگشت اشاره ما می بایست روی دکمه های J و F قرار گیرند و نیز اینکه هر کدام از انگشتان دست ما صرفاً می تواند حروف خاصی از صفحه کلید را فشار دهد به راحتی خواهیم توانست تایپ ده انگشتی را فرا گیریم. نکته ای که در فراگیری نحوه تایپ ده انگشتی می تواند خیلی به ما کمک بکند این است که در مرحله اول سعی کنیم در تمرین های خود کلماتی را بنویسیم که کلیه حروف آن در خط وسط قرار دارند مثل کلمه ask سپس کلماتی را تمرین کنیم که کلیه حروف آن در خط بالای صفحه کلید قرار دارند مثل کلمه top و در نهایت کلماتی را تمرین کنیم که تلفیقی از کلیه حروف صفحه کلید می باشند. (علت اینکه با خط پایین صفحه کلید کاری نداریم این است که به دشواری می توان کلمه ای معنا دار با استفاده از حروف خط پایین صفحه کلید نوشت). شاید در هفته های اول این کار کمی دشوار به نظر برسد و از این کار خسته شویم، اما با گذشت زمان نه تنها به این مهارت تسلط پیدا خواهیم کرد بلکه از تایپ کردن هم لذت خواهیم برد.

مهارت دومی که برای یک برنامه نویس لازم و ضروری است فراگیری زبان انگلیسی است. در واقع دلایل مختلفی وجود دارند که بنده تاکید دارم که برنامه نویسان مبتدی می بایست حداقل به زبان انگلیسی آشنایی داشته باشند. (اگر توجه کرده باشید بنده به هیچ وجه از اصطلاح تسلط به زبان انگلیسی استفاده نکردم چرا که اگر نگوئیم تسلط به یک زبان خارجی کاری محال است، می توانیم بگوئیم که کاری بسیار دشوار می باشد). از جمله این دلایل می توان گفت که یک برنامه نویس برای اینکه همواره به روز باشد و از آخرین پیشرفت ها در زمینه برنامه نویسی مطلع باشد می بایست با زبان انگلیسی آشنا باشد. از سوی دیگر Documentation یا اسناد زبان برنامه نویسی جاوا کاملاً به زبان انگلیسی نوشته شده است و برای خواندن آن ها ما می بایست با زبان انگلیسی آشنا باشیم. یکی دیگر از مواردی که دانستن زبان انگلیسی در آن به کمک ما می آید مکاتبه با دیگر

دوره آموزش جاوا

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

برنامه نویسان در سرتاسر دنیا است. به طور مثال وب سایتی به آدرس www.stackoverflow.com این امکان را به ما می دهد که سوال یا مشکل خود را به همراه کدی که نوشته ایم را به اشتراک بگذاریم و از نظر دیگر برنامه نویسان پیرامون آن آگاه شویم. به طور مثال بنده در اوایل شروع برنامه نویسی اندروید اقدام به نوشتن یک برنامه اندروید برای BMI یا فرمول محاسبه تناسب قد و وزن کردم و در اجرای کد خود با مشکل مواجه شده بودم. نکته جالب توجه اینجا بود که فقط و فقط سه دقیقه پس از مطرح کردن مشکل خود در سایت فوق الذکر چهار جواب دریافت کردم و مشکل برنامه نیز حل شد که این کار بدون آشنایی با زبان انگلیسی به هیچ وجه امکان پذیر نبود.

نظرات گوناگونی پیرامون این مسئله وجود دارد که برای درک مطالب به زبان انگلیسی به چه تعداد واژه نیاز داریم. به طور مثال در زبان انگلیسی در حدود یک میلیون واژه وجود دارد که هر روز تعدادی به این لیست اضافه می گردد. اما سوال اینجا است که یک انگلیسی زبان به چه تعداد از این واژگان برای خواندن یک کتاب به زبان انگلیسی نیاز دارد. طبق تحقیقات صورت گرفته، از این لیست واژگان که تعداد آن هم کم نیست در حدود ۲ هزار واژه وجود دارد که در درصد قابل توجهی از متون عمومی مورد استفاده قرار می گیرند. به عبارت دیگر می توان گفت که برنامه نویسان عزیز می توانند با فراگیری صرفاً ۲۰۰۰ واژه پرکاربرد زبان انگلیسی بخش قابل توجهی از مکالمات و مکاتبات خود (حدوداً ۸۵ درصد) را با دیگر برنامه نویسان دنیا انجام داده و یا به سهولت بتوانند کتاب های مرجعی که به زبان انگلیسی نوشته شده اند را مطالعه نمایند. حال ۱۵ درصد مابقی واژگان که جزو آن ۲۰۰۰ واژه پرکاربرد نیستند هم مربوط به تخصصی است که افراد بر مبنای آن، واژگان خاصی را مورد استفاده قرار می دهند. به طور مثال یک برنامه نویس دامنه خاصی از واژگان را مورد استفاده قرار می دهد که معانی آن واژگان ممکن

است برای افراد با حرفه های متفاوت نا آشنا باشد. به عنوان مثال می توان Keyword های زبان برنامه نویسی جاوا را جزو آن ۱۵ درصد باقیمانده در نظر گرفت.

پیشنهادی که به علاقمندان به برنامه نویسی جاوا می توان داد این است که دفترچه ای را تحت عنوان واژه نامه برای نوشتن اصطلاحات لاتین برنامه نویسی جاوا به همراه تعاریف آن ها اختصاص داده تا بتوانند در صورت لزوم به مرور اصطلاحات خاص این زبان پردازند.

در مقدمه آموزش برنامه نویسی جاوا ما با مفهوم برنامه نویسی کامپیوتر به طور کلی آشنا شده و از بین زبان های برنامه نویسی به ذکر تاریخچه و همچنین مزایای زبان برنامه نویسی جاوا که در این سری از آموزش ها مد نظر ما است پرداختیم. سپس محیطی که در آن بتوان کد نویسی کرد را آماده ساخته و منتظر خواهیم ماند تا در قسمت بعدی آموزش اولین برنامه جاوا خود را اجرا نماییم. در نهایت به دو ابزار بسیار مهم در حرفه برنامه نویسی اشاره شد که امیدواریم علاقمندان در کنار فراگیری زبان برنامه نویسی جاوا، این مهارت های جانبی را نیز دنبال نمایند.