خب، Java Virtual Machine یک ماشین انتزائیه که برنامه های Java رو اجرا میکنه ولی این محیط از چند بخش تشکیل شده؛ به عنوان مثال برای اجرای برنامه های جاوا ما نیاز به یک محیطی داریم که بتونیم توش برنامه های جاوامون رو اجرا کنیم که اون محیط  JRE و یا Java Runtime Environment نام داره همچنین برای توسعه برنامه های جاوا ما نیاز به JDK و یا Java Development Kit داریم که با این کیت بتونیم برنامه های جاوامون رو توسعه بدیم.

خب به قسمتی میرسیم که ببنیم که کاتلین چگونه برنامه های خودش رو اجرا میکنه ولی قبلش باید ببینیم که جاوا چجوری یک فایل رو اجرا میکنه.

اول این که ما مسلماً یک سری کد مینویسم و  توی یک فایل مثلا Customer.java ذخیره میکنیم. حالا کامپایلر جاوا رو صدا میزنیم که اون کد هارو جوری کامپایل کنه که اون فایل ها قابل اجرا باشن بر روی ماشین مجازی، پس کامپایلر یک سری بایت کد درست میکنه که اون فایل ها اجرا بشن.

حالا بریم سمت کاتلین:

فرض میکنیم که ما میخواهیم فایل Customer.kt را اجرا کنیم.ابتدا این فایل توسط کامپایلر کاتلین به فایل Customer.Class تبدیل میشه که این فایل همون ByteCode های جاواست و این فایل همون فایلیه که توسط JVM اجرا میشه. دقت کردین که اینجا هیچگونه بار اضافه اجرایی یا به اصطلاح ( overhead ) به سیستم وارد نمیشه و عملا داریم همون بایت کد هایی رو اجرا میکنیم که اگه کدمون رو به جاوا مینوشتیم درست میشد!

همون طور که گفتیم یکی از خصوصیات Kotlin اینه که باید قابلیت همکاری با Java را داشته باشد و همینطور که می بینین همینجوریه چون عملا بایت کد ها چون از یک جنس هستند میتونن با هم همکاری کنن ولی خب در بخش همکاری پذیری به این ویژگی بیشتر میپردازیم.

پس یادمون باشه که نیازی به نوشتن یا تبدیل کردن پروژه های جاوامون نیست! ما میتونیم همینجوری از اون ها داخل پروژه های جدیدمون استفاده کنیم، بعله یعنی میتونید توی یک پروژه هم کاتلین کد بزنین و هم جاوا!