\*برای تمرین بهتر است از IntelliJ IDEA استفاده شود\*

یکی از ویژگی های کاتلین Null-Safe بودن این زبونه و دیدیم که بصورت پیشفرض تایپ ها نمیتونن null باشن و اگر میخواین که یک متغیر رو null کنین باید بیان کنین که اون nullable هه ولی خب زمانی که نوبت به همکاری با Java میرسه قضیه فرق میکنه، چراکه همه چی میتونه توی جاوا null باشه. خب اینجا باید چیکار کنیم؟

راستیتش قضیه خیلی سادست. به عبارتی برمیخوریم که ممکنه null باشه،خیلی ساده تنها از عملگر “?” استفاده میکنیم و تمام. یک مشکلی اینجا پیش میاد، مشکل که نه درواقع یک جور میشه گفت کدمون زشت میشه چرا که عبارتی خواهیم داشت که در اون به صورت تفریط از "?." استفاده شده، که این باعث میشه کدمون زشت بشه. برای حل این موضوع میتونیم از Annotation ها استفاده کنیم، بگیم که ما میدونیم که این عبارت ممکن نیست که null بشن و نیازی نداریم که از عملگری استفاده کنیم که این بازم مشکلات خودشو به وجود میاره. راه حل نهایی چیه ؟ درواقع زمانی که با جاوا کار میکنیم تصمیم نهایی با توسعه دهنده است که عبارت صدا زده شده این قابلیت رو داشته باشه که بتونه null باشه یا نه.

بذارین با کد واضح تر براتون توضیح بدم. تا اینجا یادتون میاد که ما کلاسی به نام KotlinCustomRepo رو به صورت زیر ساختیم :

class KotlinCustomerRepo : CustomerRepository {  
 override fun getAll(): MutableList<CustomerJava>? {  
 *TODO*("not implemented")  
 }  
 override fun getById(id: Int): CustomerJava? {  
 *TODO*("not implemented")  
 }  
}

ولی اگه من ? رو پاک کنم بازم همه چیز درسته!

class KotlinCustomerRepo : CustomerRepository {  
 override fun getAll(): MutableList<CustomerJava> {  
 *TODO*("not implemented")  
 }  
 override fun getById(id: Int): CustomerJava {  
 *TODO*("not implemented")  
 }  
}

و این درواقع دست توسعه دهنده است که به اون عبارت قابلیت داشتن null بده یا نه.

بیاین بریم سراغ کلاس CustomerJava و دوتا متد به اون اضافه کنیم

public @NotNull String neverNull(){  
 return "A String";  
}  
  
public String someTimesNull(){  
 return "A String";  
}

برای متد اول از یکی از Annotation های IntelliJ استفاده کردیم که مشخص میکنه این عبارت null برنمیگردونه. (میتونه از Annotation های دیگه هم استفاده بشه) ولی دقیقا تفاوت این توی کاتلین چیه؟ تفاوت اینجاست که وقتی در یک کلاس کاتلین میخوایم از این توابع استفاده کنیم، کاتلین متوجه میشه که متد اول نمیتونه null باشه و این درحالیه که متد دوم رو nullable تصور میکنه.