

1

0. <https://github.com/AmirLmnop/Eid-Project.git>

1. EntityNotFoundException به صورت unchecked تعریف میشه چون از RuntimeException ارث بری میکنه.

تفاوت اکسپشن های checked و unchecked :

Checked : برای خطاهای قابل پیشبینی که برنامه نویس باید آنها را مدیریت کند.

در زمان کامپایل بررسی میشوند. باید با try-catch هندل شوند یا با throws در متود اعلام شوند

unchecked : برای خطاهای غیرمنتظره و منطقی یا اشتباهات برنامه نویسی که معمولا نیاز به هندلینگ اجباری ندارند که در زمان اجرا اتفاق می افتند و از RuntimeException ارث بری میکنند.

2. با استفاده از inheritance و type casting میتوان ورودی و خروجی زیرکلاس های Entity را تغییر داد. که به دلیل polymorphism در جاوا ممکن است.

Human : inheritance از Entity ارث بری میکنه پس هر Human یک Entity است.

type casting : اگر Entity دریافت شده از Database.get() قطعاً Human باشد میتوانیم با type casting ان را به Human تبدیل کنیم.

Polymorphism : چون Human زیرکلاس Entity است میتواند یک شی Human را به شکل Entity ذخیره کند.

3. هر تغییری روی متد روی شی اصلی اعمال میشه نه یک کپی جدید. مثلاً شی humans[0] به متد add ارسال میشه و متد مستقیماً ایدی همان شی اصلی رو تغییر میده.

1

4در ("Human new = ali Human") یک شیء Human به نام ali ایجاد می شود.
در (add.Database.ali) این شیء به دیتابیس اضافه می شود.
در این خط کد اشاره گر شیء ali به لیست entities در دیتابیس اضافه می شود نه یک کپی جدید از شیء.
کند می تغییر " "Ali Hosseini" به شیء name مقدار "Ali Hosseini" = ali.name در
از آنجا که هم متغیر ali در main و هم عنصر در لیست entities به همان شیء در حافظه اشاره می کنند، این
تغییر روی همان شیء اعمال می شود.

Human aliFromTheDatabase = (Human) Database.get(alid)

که قبلاً است. این متد همان شیء اصلی را برمی گرداند نه یک کپی جدید تغییر کرده
وقتی یک شیء به یک متد منتقل میشود یا در یک لیست ذخیره می شود، در واقع ارجاع به آن شیء منتقل
می شود، نه یک کپی جدید از آن شیء.
در نتیجه هم ali در main و هم نسخه ذخیره شده در دیتابیس به یک شیء واحد در حافظه اشاره می کنند.
هر تغییری که از طریق هر یک از این ارجاع ها انجام شود، روی همان شیء واحد اعمال می شود.