

## گزارش کار پروژه عید - برنامه سازی پیشرفته (جاوا)

### پاسخ سوالات قدم اول

پیشوا آذیز - 40313003

#### پاسخ سوال ۰

لینک ریپازیتوری گیت هاب پروژه: <https://github.com/PeshawaAziz/todo-list>

#### پاسخ سوال ۱

تفاوت checked و un-checked اکسپشن ها در نحوه هندل کردن آن هاست؛ به طوری که اکسپشن های checked حتماً باید توسط بلوک try catch هندل شوند. همچنین، متدی که احتمال رخ دادن اکسپشن checked در آن وجود داشته باشد باید توسط کلمه کلیدی throws مشخص شود. این نوع از اکسپشن ها در هنگام کامپایل کردن کد بررسی می شوند، به همین دلیل به صورت صریح نیاز به هندل شدن دارند، در حالی که اکسپشن های un-checked در هنگام اجرا شدن برنامه بررسی شده و لزوماً نیاز به هندل شدن ندارند. در اینجا، مطابق با خواسته پروژه، اکسپشن EntityNotFoundException می بایست از نوع un-checked می بود. در جاوا اکسپشن هایی که از کلاس Exception ارث بری می کنند، از نوع checked و اکسپشن هایی که از کلاس RuntimeException ارث بری می کنند از نوع un-checked هستند. لذا اکسپشن مورد نظر را زیرکلاسی از کلاس RuntimeException قرار دادم.

#### پاسخ سوال ۲

در ابتدای متد main در کلاس Main، سه آبجکت از نوع Human تعریف شده و در خط های بعدی به دیتابیس اضافه شده اند. در این صورت، چون کلاس Human از Entity ارث بری می کند، کامپایلر بدون مشکل این آبجکت ها را upcast کرده و آن ها را به صورت Entity ذخیره می کند. (Implicit casting)

حال، در صورتی که لازم باشد این Entity ها را دوباره به صورت Human ذخیره کنیم، کامپایلر بدون مشکل این downcast را انجام می‌دهد. دلیل این که اکسپشن ClassCastException در این مورد به وجود نمی‌آید این است که آبجکت‌های ذخیره شده به صورت Entity در دیتابیس در اصل به صورت رفرنس‌هایی از آبجکت‌های Human هستند و کامپایلر بدون مشکل cast ذکر شده را انجام می‌دهد.

### پاسخ سوال ۳

وقتی که آبجکت‌های Human را تعریف می‌کنیم، در اصل هر یک از آن‌ها (humans[0] و ...) یک رفرنس به آبجکتی از نوع Human در مموری است و با تغییر دادن آن، آبجکت مورد نظر در مموری دچار تغییر می‌شود. حال با اضافه کردن هر کدام از این آبجکت‌ها به دیتابیس، به دلیل این که id آن‌ها -از طریق رفرنس- آپدیت شده و تغییر می‌کند، هر جایی که از این رفرنس استفاده شده باشد، آن هم تغییر می‌کند. لذا id آبجکت humans[0] با اضافه شدن به دیتابیس، تغییر می‌کند.

### پاسخ سوال ۴

باز هم دلیل این رخداد به رفرنس‌ها و نحوه عملکرد آن‌ها در جاوا برمی‌گردد. با اضافه کردن آبجکت ali به دیتابیس، در اصل یک رفرنس به آن در مموری، در دیتابیس ذخیره می‌شود. حال هرگونه تغییر در آن، موجب تغییر در آبجکت ذخیره شده در مموری می‌شود. بعد با ایجاد یک آبجکت جدید با نام aliFromTheDatabase و فراخوانی متد Database.get، چیزی که توسط متد نام برده برگشت داده می‌شود، باز یک رفرنس به آبجکت قبلی است. لذا تغییر انجام شده در آن یعنی مقدار جدید ali.name که برابر با aliFromTheDatabase است، مشاهده می‌شود.