

## مینی پروژه 8 : مدیریت استثنا

در این مینی پروژه، بخشی از عملکرد رزرو پرواز را پیاده سازی می کنید که در آن مسافران ابتدا باید ثبت نام کرده و سپس می توانند از بین پرواز های موجود، یک یا چند پرواز را رزرو کنند.

هدف سوال : مفهوم استثنا و مدیریت آن با بلوک های try/catch، سلسله مراتب استثناها، پرتاب عمدی استثنا، کلاس Exception به عنوان پدر همه خطاها، کلمه کلیدی throws، تعریف استثنا جدید، بلوک finally.

دقت کنید برای پیاده سازی این مینی پروژه باید از مفاهیم گفته شده استفاده کنید.

### موجودیت ها:

#### ❖ پرواز

- تاریخ پرواز (روز، ماه و سال) (از نوع کلاس **LocalDate**)
- ساعت پرواز (ساعت و دقیقه) (از نوع کلاس **LocalTime**)
- (می توانید برای راحتی کار، از کلاس **LocalDateTime** استفاده کنید که هر دو مقدار زمان و تاریخ را نگه می دارد.)
- لیست مسافران
- مبدا (رشته)
- مقصد (رشته)
- مسافت (عدد اعشاری به کیلومتر)
- مدت زمان پرواز (به دقیقه)
- سرعت پرواز (برحسب متر بر ثانیه)
- هزینه سفر (عدد اعشاری)
- ظرفیت هواپیما (عدد صحیح)

#### ❖ مسافر

- نام و نام خانوادگی (رشته)
- شماره تماس (رشته)
- ایمیل (رشته)
- اعتبار حساب (عدد اعشاری)
- لیست از پروازهای رزرو شده
- قابلیت ها:
- ✓ رزرو پرواز
- ✓ نمایش رزروهای انجام شده

## استثناها:

یک پکیج به نام exceptions بسازید و تمامی کلاس های استثنا را در آن تعریف کنید.

❖ کلاس های استثنا برای اطلاعات نامعتبر: استثنای "اطلاعات نامعتبر مسافر" یک کلاس پدر است. کلاس هایی که از این استثنا ارث بری می کنند، موارد زیر می باشند:

➤ استثنای "ایمیل نامعتبر": وارد شدن ایمیل با فرمت اشتباه (بررسی فرمت با Regex) در هنگام ایجاد شی مسافر

➤ استثنای "شماره تلفن نامعتبر": شماره تلفن باید با 98 شروع شده و طول کل آن 12 رقم باشد (بررسی فرمت با Regex).

➤ استثنای "موجودی نامعتبر": وارد شدن موجودی منفی هنگام ایجاد شی مسافر

❖ کلاس استثنا برای "رزرو پرواز ناموفق": این استثنا شامل کلاس های زیر است که از آن ارث بری می کنند:

➤ استثنای "عدم موجودی کافی": پس از انتخاب بلیط مدنظر باید هزینه سفر از اعتبار حساب مسافر کسر گردد. در صورتی که موجودی حساب مسافر از هزینه پرواز کمتر باشد استثنای "عدم موجودی کافی" تولید شود. (اگر ثبت بلیط موفقیت آمیز نباشد و موجودی ناکافی باشد، چیزی از موجودی کاسته نمی شود)

➤ استثنای "تداخل زمانی": مسافر می تواند امکان ثبت چند بلیط را داشته باشد. به این منظور، باید چک شود سفرها تداخل نداشته باشند. یعنی فاصله اتمام پرواز قبلی تا شروع پرواز بعدی حداقل دو ساعت باشد. (نیازی نیست حالتی که به روز بعد منتقل می شویم را در نظر بگیرید).

➤ استثنای "تداخل مبدا و مقصد پروازها": مقصد یک پرواز و مبدا پرواز بعدی باید یکسان باشد در غیر این صورت باید این استثنا ایجاد شود.

❖ نکته: هرکدام از استثنای ذکر شده در بالا، یک کلاس هستند که از کلاس Exception ارث بری می کنند. در این کلاس ها، پیام خطای مناسب برای هر استثنا را به سازنده پدر آن (super) بدهید تا هنگام فراخوانی متد getMessage، آن پیام را چاپ کند.

❖ نکته: برای تمامی کلاس های استثنا که ذکر شد، باید یک سازنده بدون آرگومان استرینگ و یک سازنده با آرگومان استرینگ پیاده سازی شود. در حالتی که سازنده یک مقدار String را دریافت می کند، پیام چاپ شده برای این استثنا باید حاوی پیام کلاس پدرش + خود آن String باشد. در حالتی که سازنده بدون آرگومان است، باید یک پیام پیش فرض برای آن در نظر بگیرید و پیام مربوط به آن استثنا، باز هم ترکیبی از پیام کلاس پدر و آن پیام باشد. مثلاً برای استثنای "تداخل زمانی پرواز" پیام خطای آن باید به این صورت باشد: "رزرو پرواز ناموفق - تداخل زمانی پروازها". (برای این کار باید در هر کلاس پدر، رشته پیام همان کلاس را به رشته دریافتی از آرگومان های سازنده بچسبانید).

❖ نکته: استثنایهای مربوط به ثبت نام در توابعی که با کلمه `check` شروع می شوند پرتاب می شوند و استثنایهای مربوط به رزرو پرواز ناموفق در متد `reserveFlight` پرتاب می شوند.

## توضیحات کلی مسافر و رزرو پرواز:

- ثبت پرواز چه موفقیت آمیز باشد چه نباشد، موجودی کاربر و اطلاعات پرواز مورد نظر نمایش داده شود. دقت کنید برای این قسمت باید از یکی از مفاهیم این مبحث استفاده کنید. بدین صورت که در هنگام مدیریت استثنا، هم در صورت پرتاب استثنا و هم در صورت پرتاب نشدن آن، اطلاعات خواسته شده نمایش داده شود.
- برای ثبت سرعت، در سازنده از مسافت و زمان پرواز استفاده کنید. در صورت مشکل در محاسبات (تقسیم بر صفر) خطای مربوطه مدیریت گردد.
- در صورتی که ظرفیت پرواز تکمیل شده بود، باید استثنای `IndexOutOfBounds` پرتاب شود. دقت کنید نیازی به نوشتن کلاس برای این استثنا نیست؛)
- در هر قسمتی از برنامه که ممکن است خطای `"Index Out Of Bounds"` یا `"Input Mismatch"` رخ دهد، باید آن را مدیریت کنید. (استثنای اول هنگام استفاده از اندیس های غیرمجاز آرایه و لیست و استثنای دوم هنگامی رخ می دهد که به اسکنری از یک نوع مشخص، ورودی ای با نوع دیگر بدهید. مثلا وارد کردن رشته یا کاراکتر بجای عدد صحیح).

## توضیحات کلی:

- پکیج بندی مناسب باید رعایت شود. کلاس های پرواز و مسافر باید در یک پکیج و کلاس های مربوط به استثناها در یک پکیج باشند. همچنین کلاس `Main` که در آن برنامه را تست کرده اید، باید خارج از این دو پکیج قرار بگیرد. (دقت کنید که در کد نمونه ای داده شده، پکیج بندی رعایت نشده است و خودتان باید آنها را ایجاد کنید).
- برای کلاس های برنامه متد `toString` را `Override` کنید و هنگام نمایش اطلاعات اشیا، از آن استفاده کنید.
- خطاها باید به طور کامل در `main` مدیریت (با استفاده از `try/catch` های جدا برای هر خطا) شوند. (توجه کنید که همه استثنایهای ذکر شده در بالا، نهایتا باید در مین مدیریت شوند و تست آنها به گونه ای باشد که اجرای برنامه متوقف نشود).
- در بخش هایی از تست برنامه در `main` که قرار است متدهای مربوط به ساخت شی مسافر آزمایش شوند، از کلاس استثنای پدر (استثنای "اطلاعات نامعتبر") برای `catch` استثناها استفاده کنید.
- برای مدیریت استثنایهایی که از "رزرو پرواز ناموفق" ارث بری می کنند و در قسمت `catch` عملیات یکسانی انجام می شود، از یک بلوک `catch` برای مدیریت آنها استفاده کنید. (با استفاده از `|`)
- **نکته مهم:** هنگام ارائه، از شما خواسته می شود بخش هایی از تست کد (از جمله `try` و `catch` ها و همچنین حالت های مختلف برای پرتاب استثنا) را تغییر دهید تا ایجاد و مدیریت خطاهای ذکر شده در بالا بررسی شوند. متناسب با نحوه پاسخگویی شما، عددی بین 0 تا 1 به عملکرد شما داده می شود که به عنوان ضریب تسلط در نمره شما ضرب می شود.

