

بسمه تعالی

## تکلیف مهندسی نرم افزار پیشرفته

موعد تحویل ۱۴۰۴/۱۰/۰۷

در این تکلیف قصد داریم با مفاهیم بیان شده در فصل آزمون نرم افزار بیشتر آشنا شویم. بدین منظور پروژه‌ای که با هدف رزرو قطار پیاده سازی شده است را در نظر بگیرید. این پروژه یک کتابخانه کوچک تحت جاوا فراهم می کند که از آن می توان برای ساخت یک سیستم ساده رزرو قطار استفاده کرد. این کتابخانه بخش هایی برای تعریف قطار، شهر، سفر و بلیط فراهم می نماید. این بخش ها باید اطمینان حاصل نمایند که در سیستم نهایی، سفرها با یکدیگر سازگار هستند (مثلاً هم پوشانی و تلافی ندارند) و امکانات مدیریت (مانند راهی برای به تأخیر انداختن سفر) فراهم شده است. کدهای جاوای مربوط به این پروژه در پوشه پیوست شده به فایل تکلیف قرار گرفته اند، اما خلاصه ای از نحوه استفاده و اجرای آن به شرح زیر می باشد.

- برای کامپایل و اجرا دستور زیر را بنویسید:

```
mvn package
```

- به منظور استفاده از کدهای بخش های مختلف اطلاعات زیر را در نظر بگیرید:
- ✓ برای ایجاد یک نمونه از سیستم باید یک شیء از نوع TicketReservationSystem بسازید که نیاز به یک نمونه از کلاس TicketReservationSystemImpl دارد و آن نیز نیاز به انتخاب منطقه زمانی دارد. بدین منظور کدهای زیر را در نظر بگیرید:

```
// Choosing the time zone of the system, to parse times/dates
// Here we use the time zone of the system
ZoneId timeZone = ZoneId.systemDefault();

// Creating the system
TicketReservationSystem system = new TicketReservationSystemImpl(timeZone);
```

✓ بعد از ایجاد سیستم، بایستی اطلاعات اولیه مربوط به شهرها و قطارها و ... را مانند کد زیر ایجاد کنید:

```
// Creating and adding cities in the system
City rennesCity = new CityImpl("Rennes");
City nantesCity = new CityImpl("Nantes");
system.addCity(rennesCity);
system.addCity(nantesCity);

// Creating and adding trains in the system
Train smallTrain = new TrainImpl("Small Train", 2);
Train biggerTrain = new TrainImpl("Bigger Train", 10);
system.addTrain(smallTrain);
system.addTrain(biggerTrain);
```

✓ حال نوبت ایجاد سفر است، برای این کار بایستی همانند کد زیر تاریخ ورود و خروج و دیگر اطلاعات مربوط به سفر را تعریف کنید:

```
Instant departure1 = TimeManagement.createInstant("2022-05-12 12:00", timeZone);
Instant arrival1 = TimeManagement.createInstant("2022-05-12 13:00", timeZone);
Instant departure2 = TimeManagement.createInstant("2022-05-12 15:00", timeZone);
Instant arrival2 = TimeManagement.createInstant("2022-05-12 16:00", timeZone);
try {
    Trip trip1 = system.createTrip(rennesCity, nantesCity, smallTrain, departure1, arrival1);
    Trip trip2 = system.createTrip(rennesCity, nantesCity, biggerTrain, departure2, arrival2);
} catch (TripException e) {
    // Do something if a constraint is unfulfilled
}
```

✓ برای رزرو کردن بلیط همانند کد زیر عمل کنید:

```
try {
    Ticket ticket1 = trip1.bookTicket("Alice");
    Ticket ticket2 = trip1.bookTicket("Bob");
} catch (ReservationException e) {
    // Do something if reservation was not possible
}
```

✓ برای لغو کردن بلیط دستور زیر را در نظر بگیرید:

```
trip1.cancel(ticket1);
```

✓ برای تغییر بلیط نیز کد زیر را در نظر داشته باشید:

```
List<Trip> alternatives = system.findPossibleExchanges(ticket2);
try {
    ticket2.exchangeTicket(alternatives.get(0));
} catch (ReservationException e) {
    // Do something if the exchange is not possible
}
```

✓ همچنین برای یافتن اطلاعات سفرهای موجود، می‌توانید از کد زیر استفاده کنید:

```
// Search all trips from a city, on a given day
List<Trip> results1 = system.findAvailableTrips(rennesCity, LocalDate.of(2022, 5, 12));

// Search only trip between two cities, on a given day
List<Trip> results2 = system.findAvailableTrips(rennesCity, nantesCity, LocalDate.of(2022, 5, 12));
```

✓ برای لغو کردن یک سفر از دستور زیر استفاده کنید:

```
system.cancelTrip(trip1);
```

✓ در نهایت برای به تأخیر انداخت سفر نیز کد زیر را در نظر داشته باشید:

```
// Delay departure
system.delayTripDeparture(trip2, Duration.ofHours(1));

// Delay arrival
system.delayTripArrival(trip2, Duration.ofMinutes(30));
```

❖ حال با توجه به این توضیحات شما بایستی،

✓ در بخش اول این تکلیف، با دیدگاه تست جعبه سیاه و بخش‌بندی پارامترهای ورودی، **تحلیل**

**و محاسبه** نمایید تعداد کل تست‌های واحد مورد نیاز چند عدد است؟

✓ در بخش دوم تکلیف بایستی ۵ سناریو و مورد آزمون (Test Case) برای بررسی صحت

عملکرد کدهای پروژه‌ای که با هدف رزرو قطار پیاده‌سازی شده است، **طراحی** نمایید.

✓ در بخش سوم این تکلیف، ۵ سناریو بر اساس ساختار Gherkin برای آزمون پذیرش سیستم نهایی به گونه‌ای طراحی نمایید که اصلی‌ترین نیازهای بیان شده در متن تکلیف را پوشش دهد.

❖ حال با توجه به کدهای موجود در پروژه پیوست و توضیحات بیان شده در بخش‌های قبل شما بایستی،  
✓ در بخش چهارم این تکلیف، با استفاده از JUnit 5 یک کلاس تست شامل Fixture های مورد نیاز و ۱۰ عدد Test Method بر اساس سناریوهای بیان شده در بخش دوم تکلیف پیاده‌سازی نمایید.

✓ در بخش پنجم این تکلیف، با استفاده از Cucumber، ۳ مورد Step definition بر اساس سناریوهای بیان شده در بخش سوم تکلیف پیاده‌سازی نمایید.  
✓ در بخش ششم این تکلیف، شما بایستی از Mockito استفاده نموده و ۲ عدد Test Method جدید با استفاده از Mock کردن پیاده‌سازی نمایید.

#### توجه نمایید:

✓ هر بخش این تکلیف ۰.۵ نمره داشته و تکلیف بایستی به صورت فردی انجام شد.  
✓ به عنوان خروجی بخش‌های اول تا سوم، گزارشی متنی شامل توضیحات و تعداد کل تست‌های مورد نیاز بخش اول و سناریوهای بخش‌های دوم و سوم تهیه و ارسال نمایید.  
✓ ویدئویی کوتاه از اجرای تست‌های مربوط به بخش‌های چهارم تا ششم ایجاد نموده و لینک آن را در ابتدای فایل PDF پاسخ ارسالی تکلیف خود قرار دهید.

#### خروجی:

✓ گزارش نهایی را در قالب یک فایل PDF در سامانه LMS ارسال نمایید.

موفق باشید

شعرباف