

Quby Java Programming Challenge

About The Challenge

Quby dry-cleaning company is known for its speedy service. It guarantees to dry-clean anything within two business hours or less.

The problem is, when the customer drops off the clothes, he needs to know what time they are guaranteed to be done.

The Assignment

It is your job to write a Java program that will determine the guaranteed time given a business hour schedule. You must at least create a class called `BusinessHourCalculator` that allows one to define the opening and closing time for each day. By default the store is in business 7 days a week. You can use additional libraries as long as you bundle them.

The class should at least provide the following API:

You must be able to instantiate the class using the following constructor:

```
BusinessHourCalculator(String defaultOpeningTime, String defaultClosingTime);
```

The two parameters should be in a 24h time format, such as "19:00".

The instance allows for further customization of the opening hours by specifying the day of the week or a specific date. You should define an enum for the days of the week.

```
businessHourCalculator.setOpeningHours(DayOfWeek.MONDAY, "9:00", "17:00");
```

And for a specific date:

```
businessHourCalculator.setOpeningHours("2010-09-12", "9:00", "17:00");
```

You must also be able to specify that the store is closed for a specific day of week or date.

```
businessHourCalculator.setClosed(DayOfWeek.MONDAY);  
businessHourCalculator.setClosed("2010-12-25");
```

The `setClosed` method accepts multiple arguments.

```
businessHourCalculator.setClosed(DayOfWeek.MONDAY, DayOfWeek.FRIDAY);
businessHourCalculator.setClosed("2010-12-25", "2011-01-01", "2011-02-06");
```

The `setOpeningHours` method should change the opening and closing time for a given day. The `setClosed` method should specify which days the shop is not open. Notice days can either be an enum for week days or a string for specific dates. Any given day can only have one opening time and one closing time — there are no off-hours in the middle of the day.

A method called `calculateDeadline` should determine the resulting business time given a time interval (in seconds) along with a starting datetime (as a string). The returned object should be an instance of `java.util.Date`.

Examples

Given the following configuration:

```
businessHourCalculator = new businessHourCalculator("09:00", "15:00");
businessHourCalculator.setOpeningHours(DayOfWeek.FRIDAY, "10:00", "17:00");
businessHourCalculator.setOpeningHours("2010-12-24", "8:00", "13:00");
businessHourCalculator.setClosed(DayOfWeek.SUNDAY, DayOfWeek.WEDNESDAY);
businessHourCalculator.setClosed("2010-12-25");
```

the business hour calculator will yield the following results:

```
// example #1
businessHourCalculator.calculateDeadline(2*60*60, "2010-06-07 09:10");
// => Mon Jun 07 11:10:00 2010

// example #2
businessHourCalculator.calculateDeadline(15*60, "2010-06-08 14:48");
// => Thu Jun 10 09:03:00 2010

// example #3
businessHourCalculator.calculateDeadline(7*60*60, "2010-12-24 6:45");
// => Mon Dec 27 11:00:00 2010
```

In the first example the time interval is 2 hours (7,200 seconds). Since the 2 hours fall within business hours the day does not change and the interval is simply added to the starting time.

In the second example an interval of 15 minutes (900 seconds) is used. The starting time is 12 minutes before closing time which leaves 3 minutes remaining to be added to the next business day. The next day is Wednesday and therefore closed, so the resulting time is 3 minutes after opening on the following day.

The last example is 7 hours (25200 seconds) which starts before opening on Dec 24th. There are only 5 business hours on Dec 24th which leaves 2 hours remaining for the next business day. The next two days are closed (Dec 25th and Sunday) therefore the deadline is not until 2 hours after opening on Dec 27th.