



Siddharth Basu

March 20, 2020

Kata Assignment

This kata simulates a babysitter working and getting paid for one night. The rules are pretty straight forward.

Feature

As a babysitter

In order to get paid for 1 night of work

I want to calculate my nightly charge

Requirements

The babysitter:

- starts no earlier than 5:00PM
- leaves no later than 4:00AM
- only babysits for one family per night
- gets paid for full hours (no fractional hours)
- should be prevented from mistakes when entering times (e.g. end time before start time, or outside of allowable work hours)

The job:

- Pays different rates for each family (based on bedtimes, kids and pets, etc...)
- Family A pays \$15 per hour before 11pm, and \$20 per hour the rest of the night
- Family B pays \$12 per hour before 10pm, \$8 between 10 and 12, and \$16 the rest of the night
- Family C pays \$21 per hour before 9pm, then \$15 the rest of the night
- The time ranges are the same as the babysitter (5pm through 4am)

Deliverable:

- Calculate total pay, based on babysitter start and end time, and a family.



Table of Contents

Subject	Page Number
Kata Assignment	1
Table of Contents	2
Application Details	3: i
Dependencies	3: ii
Setup & Configuration	3: iii
Approach/Solution	3: iv
Project Structure	4: v
Build & Run	5: vi
Testing	6: vii



i) Application Details

The application is Java maven build. I used the JetBrains IntelliJ IDE to develop this project. The application is broken up in two sections in the src folder: main and test folders. The main contains the MainApp.java which contains the main class to run the application. I isolated the functional classes in a package called HelperClasses. There are two files: FamilyTypes.java and SitterScheduleTime.java. The test folder contains two test classes

ii) Dependencies

- Java 13
- Maven Build
- JUnit 4.12
- Hamcrest-core 1.3

iii) Setup & Configuration

- IntelliJ IDE
- Java 13 JDK Installed
- Refresh Maven Build to install dependent web jars
- Build Project
- Run MainApp.main()

iv) Algorithmic Approach/Solution

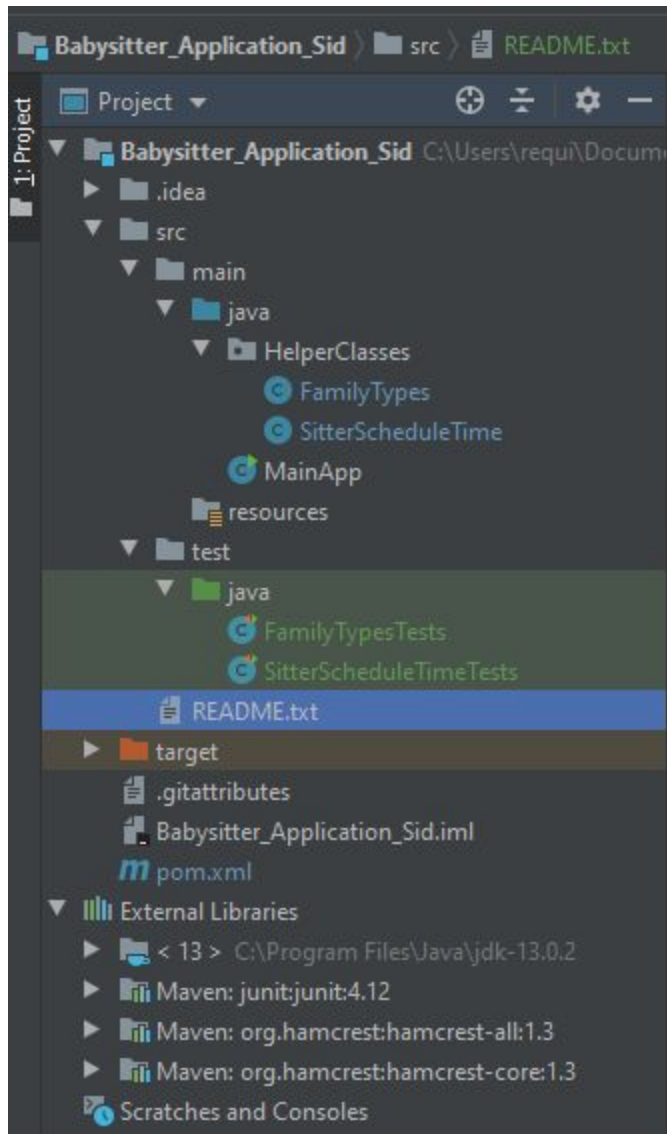
For this solution, I compartilized two models: FamilyTypes and SitterScheduleTime. These classes are being called to the MainApp where we will be running the application and the user will be asked a series of questions to calculate their total time sitting. After calculating the time worked, we give options of different Family Types (A, B, and C) to calculate the earnings for the hours worked for that family.

The FamilyTypes was to create calculation methods for the earning rates for each family type. Since each family has their own time requirements and rates associated with the times, I calculated the total number of minutes the sitter has worked at what incremented time slots, add the earnings from each slot, and then give a total minutes and earnings for the given babysitting session.

Next was the SitterScheduleTime which was to initialize parameters on the babysitter's availability for specified time. We are given some conditionals like: Sitter can work from 5PM to 4AM, Sitter must work at least 1 hour, No partial hour payments, Work for a single family a night, and users should be prevented from making mistakes while entering times.



v) Project Structure



This is a Maven build Java application. The dependencies are stored in the pom.xml which will specify the web jars required to run the application. This is a highly scalable application and we can further develop future features like connecting databases, add a web view, and more babysitter related tasks. If we want more jars or API dependencies, we can easily update the pom file and our dependencies should be pulled from the web. This makes moving the code around simpler since the developer won't need to configure anything themselves. As long as they have Java and a maven environment, you will be able to run this application almost anywhere with similar platform requirements.



vi) Build & Run

After loading up the project, locate the MainApp class and ensure your project is building properly.



Below is what a successful run will look like. You will be asked to enter your name.

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.3.4\lib\idea_rt.jar-56119-C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.3.4\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\requi\Documents\GitHub\Babysitter_Application_Sid\target\classes MainApp
Welcome to the Babysitter Application
Powered By Accenture Kata Assignment

Instructions: Enter your name. Enter starting hour, minute and AM/PM. Enter ending hour, minute and AM/PM

You will given (3) family types to choose from (A, B, C) which will let us calculate your earnings.
*-----*

What is your name?
Process finished with exit code -1
```

In IntelliJ, there is a green triangle to run application from the MainApp.main()

```
Run: MainApp x
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" ...
Welcome to the Babysitter Application
Powered By Accenture Kata Assignment

Instructions: Enter your name. Enter starting hour, minute and AM/PM. Enter ending hour, minute and AM/PM

You will given (3) family types to choose from (A, B, C) which will let us calculate your earnings.
*-----*

What is your name? Sid

What hour did you start babysitting? 5

Was it PM or AM (enter '0' for PM and '1' for AM)? 0

Starting Hour is 5
What minute did you start babysitting? 0

Starting minute is 00

What hour did you end babysitting? 4

Was it PM or AM (enter '0' for PM and '1' for AM)? 1

Ending Hour is 16

What minute did you end babysitting? 0

Ending minute is 00

The Total Minutes are 660

Which family type did you babysit for? Please enter 'A' for Type A,
'B' for the Type B or 'C' for the Type C:
a
Sid, you babysat for 660 minutes and earned $190.00 babysitting for Family A.
b
Sid, you babysat for 660 minutes and earned $140.00 babysitting for Family B.
c
Sid, you babysat for 660 minutes and earned $189.00 babysitting for Family C.
|
```



vii) Testing

-FamilyTypesTests

(27/27 Tests Passed)

Run:	MainApp	FamilyTypesTests
Run	✓	3 ms
✓	typeBIsBeginHourAfterMidnight	2 ms
✓	typeBTotalPayBeginningAtMidnight	0 ms
✓	typeCTotalPayBeginBefore9PM	0 ms
✓	familiesDefaultConstructorFunctional	0 ms
✓	typeBTotalPayBeginBefore10PM	0 ms
✓	typeBIsBeginHourBefore10PM	0 ms
✓	typeCTotalPayBeginningAfter9PM	0 ms
✓	typeBTotalPayBeginAt5PM	0 ms
✓	typeCIsBeginHourBefore9PM	0 ms
✓	typeBTotalPayBeginAfterMidnight	0 ms
✓	typeBTotalPayBeginAfter10PMAndBeforeMidnightAndEndingIn0Minutes	0 ms
✓	typeCStringsToIntegers	0 ms
✓	typeATotalPayBeginningAt5PM	0 ms
✓	typeBPayBefore10PM	0 ms
✓	typeBIsBeginHourBeforeMidnight	0 ms
✓	typeAPayBefore11PM	0 ms
✓	typeATotalPayBeginningBefore11PM	0 ms
✓	typeBStringsToIntegers	1 ms
✓	typeATotalPayBeginningAfter11PM	0 ms
✓	typeBTotalPayBeginAt10PM	0 ms
✓	typeAIsBeginningHourBefore11PM	0 ms
✓	typeBTotalPayBeginAfter10PMAndBeforeMidnight	0 ms
✓	typeCPayBefore9PM	0 ms
✓	typeAStringsToIntegers	0 ms
✓	typeBTotalPayBeginBefore10PMAndEndingIn0Minutes	0 ms
✓	typeCTotalPayBeginAt5PM	0 ms
✓	typeCTotalPayBeginAt9PM	0 ms

-SitterScheduleTimeTests

(24/24 Tests Passed)

Run:	MainApp	SitterScheduleTimeTests
Run	✓	2 ms
✓	verifySittersBeginningHourWithErrorForPMOrAM	1 ms
✓	verifySittersEndingTimeIsLessThan60MinutesBeforeBeginTime	0 ms
✓	verifySittersBeginningHour5PMOrLater	0 ms
✓	verifySittersEndingTimeIsBeforeBeginTime	0 ms
✓	verifySittersBeginningHourBefore4AMOrEarlier	0 ms
✓	verifySittersEndingHour4AMOrEarlier	0 ms
✓	verifyTheBabysittersName	0 ms
✓	verifySittersEndingHourCannotBeBefore5PM	1 ms
✓	verifySittersEndingMinuteWithOver59MinutesError	0 ms
✓	verifySittersEndingHourWithErrorForPMOrAM	0 ms
✓	verifySittersEndingMinute	0 ms
✓	verifySittersBeginningMinuteWithOver59MinutesError	0 ms
✓	verifySittersEndingHourWhenAnErrorForHoursANegativeInteger	0 ms
✓	verifySittersBeginningHourWhenAnErrorForHoursGreaterThan11PM	0 ms
✓	verifySittersBeginningHourCannotBeAfter4AM	0 ms
✓	verifySittersEndingMinuteWithNegativeMinutesError	0 ms
✓	verifySittersEndingTimeIsSameAsBeginTime	0 ms
✓	verifySittersBeginningMinute	0 ms
✓	verifySittersBeginningHourCannotBeBefore5PM	0 ms
✓	verifySittersBeginningHourWhenAnErrorForHoursANegativeInteger	0 ms
✓	verifySittersEndingHourWhenAnErrorForHoursGreaterThan11AM	0 ms
✓	verifySittersEndingHourCannotBeAfter4AM	0 ms
✓	verifySittersBeginningMinuteWithNegativeMinutesError	0 ms
✓	verifySittersEndingHourBeforeMidnightOrEarlier	0 ms
✓	verifySittersEndingTimeIsNotBeforeStartingTime	0 ms



Github Repository

https://github.com/siddharth-basu/Assignment_BabySitter_Application