Day Out Travel Service

**Grade settings**: Maximum grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes **Maximum execution time**: 120 s **Maximum memory used**: 1.50 GiB **Maximum execution file size**: 128 MiB

**Day Out Travel Service – Trip Charge Calculation**

[**Click here to download the Code Skeleton**](https://cognizant.tekstac.com/pluginfile.php/19295/mod_vpl/intro/DayOutTravelService.zip)

Day out Travel Service is one of the most widely used travel agency in the state. They are in need of a system which can be configured easily to adapt to the changes they bring in with the charges for the distance in kilometers and need to calculate the trip charge for their customers. You being their software consultant have been approached by them to develop a software system which can be used for managing their business.

**Service:** Calculating Trip Charge for each customer

**Travel**class with the below **private attributes**is provided as a part of code skeleton

|  |  |
| --- | --- |
| tripId | String |
| distanceTravelled | int |
| additionalCharge | AdditionalChargeInfo |
| chargePerKm | float |

**Getter and setter** methods for all the above attributes are provided as a part of code skeleton. Use appropriate spring annotation above the class to denote the class as component.

The value for **chargePerKm** is populated in **charges.properties** file as key=value pair. Fetch the values from property file and assign to the private attribute **chargePerKm** in Travel class using appropriate spring annotation.

**AdditionalChargeInfo** class with the below **private attribute**is provided as a part of code skeleton

|  |  |
| --- | --- |
| areaServiceCharge | Map<String,Float> |

**Getter and setter** method for the above attribute is provided as a part of code skeleton. Use appropriate spring annotation above the class to denote the class as component.

The Map should be configured using appropriate spring annotation with the below **key-values.**

|  |  |
| --- | --- |
| **Key HRA Type (String)** | **Value ServiceCharge (float)** |
| Rural | 100.0 |
| Urban | 300.0 |
| Metropolitan | 500.0 |

**Note:** Only for these **HRA classifications** additional charge needs to be **added** with the **tripCharge**. The keys are case sensitive.

**AdditionalChargeInfo** is used for only particular property, so **AdditionalChargeInfo object** should be autowired above the property via annotations in the Travel class.

Create a class called **ApplicationConfig** that has the required annotations for **scanning** and **registering** the bean definitions.

**Overview of Service:**

Write a method **public double calculateTripCharge (Travel travelObj, String hraType)**in **TravelBO** class that accepts a string and Travel Object. The string contains the type of HRA classifications. Based on the hraType, additional charge needs to be added with the trip charge and return the tripCharge. If the type of HRA classifications is not available in the Map then there is no additional charge.

**Formula to calculate total tripCharge is below:**

If the type of HRA classification is available in the map then calculation should be as follows,

**tripCharge = distanceTravelled \* chargePerKm;**

**tripCharge = tripCharge + appropriateAddtionalCharge;**

If the type of HRA classification is not available in the map then calculation should be as follows,

**tripCharge = distanceTravelled \* chargePerKm;**

**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| TravelService | calculateTripCharge | String tripId,  int distanceTravelled,  String hraType | double - tripCharge | InvalidDistanceTravelledException  **This Exception to be caught and thrown back to Main class** |
| TravelBO | calculateTripCharge | Travel travelObj, String hraType | double - tripCharge |  |
| ApplicationConfig | Contains all the configurations related to Service |  |  |  |

Create a class called **Driver** with the main method and get the inputs like tripId, distance travelled by the customer and type of HRA classification from the user. Get the **TravelService** class object by loading **ApplicationConfig** class and invoke the method **calculateTripCharge (tripId,** **distanceTravelled, hraType)** which is in the **TravelService** class to perform the implementation. Display the total Trip charge which is returned from **calculateTripCharge method in TravelBO class.**

**Business Rules & Validations:**

In **TravelService class**included the following private attribute. Use appropriate spring annotation above the class to denote the class as component.

**private TravelBO travelBoObj;**

**Getter and setter** method for the above attribute is provided as a part of code skeleton. The **TravelBO**object should be autowired above the property via annotations.

In this **TravelService class**, the method**public double calculateTripCharge (String tripId, int distanceTravelled, String hraType)**accepts tripId, distanceTravelled and hraType as the arguments. Validate the distanceTravelled, if the distance travelled is within the range, get the travel object and set the distanceTravelled and tripId in that object.

In case the distanceTravelled is not within the range, a user-defined Exception **InvalidDistanceTravelledException**should be thrown with message “**Distance Travelled is not valid**”.

Distance travelledshould be**greater than zero (4)**and**less than four hundred (400).**

If distance travelled is within the given range, call the method **calculateTripCharge (travelObj, hraType) in TravelBO class** and perform the implementation.

**Limitations and Constraints**

1.      **Travel**and**AddtionalChargeInfo class**should be in**com.spring.model**package.

2.      **InvalidDistanceTravelledException** class should be in **com.spring.exception** package.

3.      **TravelService**class should be in**com.spring.service**package**.**

4.      **TravelBO class**should be in**com.spring.bo**package.

5.      **Driver class**should be in **com.spring.main**package.

6.      **ApplicationConfig**class should be in **com.spring.config**package.

7.      All of the above mentioned java classes to be configured as **component**class using appropriate spring annotation.

8.      **TravelBO** should be injected via annotation inside **TravelService** class.

9.      The **additional charge** for each HRA classifications should be declared as a MAP and should be injected using @Value annotation.

10.  **charges.properties** file will be provided with the value for charge per Km by using @PropertySource annotation in ApplicationConfig class fetch the value.

11.  **DO NOT change the values in charges.properties file.**

**Sample Input Output 1:**

Enter the trip ID:

**URB101**

Enter the total distance travelled:

**200**

Enter the type of HRA classification:

**Urban  // HRA type is available in the Map hence additional charge is Applicable**

Total trip Charge: 2300.0

**Sample Input Output 2:**

Enter the trip ID:

**VIL101**

Enter the total distance travelled:

**50**

Enter the type of HRA classification:

**Village  // HRA type is not available in the Map hence additional charge is not Applicable**

Total trip Charge: 500.0

**Sample Input Output 3:**

Enter the trip ID:

**MEP101**

Enter the total distance travelled:

**422 // Distance travelled is not in given range**

Enter the type of HRA classification:

**Rural**

Distance Travelled is not valid

### **Automatic evaluation**[**[-]**](javascript:void(0);)

**Proposed grade: 97.5 / 100**  
**Result Description**  
Fail 1 --FunctionalTest::Could not load ApplicationConfig. Check whether you have correctly configured  
[[+]](javascript:void(0);)**Grading and Feedback**

#### DayOutTravelService/pom.xml

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

2 <modelVersion>4.0.0</modelVersion>

3 <groupId>DayOutTravelService</groupId>

4 <artifactId>DayOutTravelService</artifactId>

5 <version>0.0.1-SNAPSHOT</version>

6

7 <dependencies>

8

9 <dependency>

10 <groupId>org.springframework</groupId>

11 <artifactId>spring-context</artifactId>

12 <version>4.3.10.RELEASE</version>

13 </dependency>

14

15 </dependencies>

16

17 </project>

#### DayOutTravelService/src/main/java/com/spring/bo/TravelBO.java

1 *package* com.spring.bo;

2 *import* java.util.Map;

3

4 *import* com.spring.model.Travel;

5 *import* org.springframework.stereotype.Component;

6 @Component

7 *public* *class* TravelBO {

8

9 *public* *double* calculateTripCharge(Travel cObj,String hraType) {

10

11 *double* tripCharge=0.0;

12

13 Map<String,Float> map = cObj.getAdditionalCharge().getAreaServiceCharge();

14 *if*(map.containsKey(hraType))

15 {

16 *return* map.get(hraType)+cObj.getChargePerKm()\*cObj.getDistanceTravelled();

17 }

18 tripCharge = cObj.getChargePerKm()\*cObj.getDistanceTravelled();

19 *return* tripCharge;

20 }

21

22 }

23

#### DayOutTravelService/src/main/java/com/spring/config/ApplicationConfig.java

1 *package* com.spring.config;

2

3 *import* org.springframework.context.annotation.Bean;

4 *import* org.springframework.context.annotation.ComponentScan;

5 *import* org.springframework.context.annotation.Configuration;

6 *import* org.springframework.context.annotation.PropertySource;

7

8 *import* com.spring.bo.TravelBO;

9 *import* com.spring.model.AdditionalChargeInfo;

10 *import* com.spring.model.Travel;

11 *import* com.spring.service.TravelService;

12

13 @ComponentScan

14 @Configuration

15 @PropertySource("classpath:charges.properties")

16

17 *public* *class* ApplicationConfig {

18 @Bean

19 *public* Travel travelBean(){

20 Travel t = *new* Travel();

21 *return* t;

22 }

23

24 @Bean

25 *public* AdditionalChargeInfo addBean(){

26 AdditionalChargeInfo aci = *new* AdditionalChargeInfo();

27 *return* aci;

28 }

29

30 @Bean

31 *public* TravelService travelServiceBean(){

32 TravelService ts = *new* TravelService();

33 *return* ts;

34 }

35 @Bean

36 *public* TravelBO travelBoBean(){

37 TravelBO tb = *new* TravelBO();

38 *return* tb;

39 }

40 }

41

#### DayOutTravelService/src/main/java/com/spring/exception/InvalidDistanceTravelledException.java

1 *package* com.spring.exception;

2

3 *public* *class* InvalidDistanceTravelledException *extends* Exception {

4

5 *public* InvalidDistanceTravelledException(String msg) {

6 *super*(msg);

7 }

8

9 }

10

#### DayOutTravelService/src/main/java/com/spring/main/Driver.java

1 *package* com.spring.main;

2

3 *import* java.util.Scanner;

4

5 *import* org.springframework.context.ApplicationContext;

6 *import* org.springframework.context.annotation.AnnotationConfigApplicationContext;

7

8 *import* com.spring.config.ApplicationConfig;

9 *import* com.spring.exception.InvalidDistanceTravelledException;

10 *import* com.spring.service.TravelService;

11

12 *public* *class* Driver {

13

14 *public* *static* *void* main(String[] args) {

15

16 Scanner sc = *new* Scanner(System.in);

17

18 ApplicationContext context = *new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

19 TravelService ts = context.getBean(TravelService.*class*);

20

21 System.out.println("Enter the trip ID:");

22 String id = sc.next();

23

24 System.out.println("Enter the total distance travelled:");

25 *int* dist = sc.nextInt();

26

27 System.out.println("Enter the type of HRA classification:");

28 String hraType = sc.next();

29

30 *try*{

31 *double* trip = ts.calculateTripCharge(id, dist, hraType);

32 System.out.println("Total trip charges :"+trip);

33 }

34 *catch*(InvalidDistanceTravelledException e)

35 {

36 System.out.println(e.getMessage());

37 }

38

39 }

40

41 }

42

#### DayOutTravelService/src/main/java/com/spring/model/AdditionalChargeInfo.java

1 *package* com.spring.model;

2

3 *import* java.util.Map;

4 *import* org.springframework.beans.factory.annotation.Value;

5 *import* org.springframework.stereotype.Component;

6

7 @Component

8 *public* *class* AdditionalChargeInfo {

9

10 @Value("#{'Rural':100.0,'Urban':300.0,'MEtropolitan':500.0}")

11 *private* Map<String,Float> areaServiceCharge;

12

13 *public* Map<String, Float> getAreaServiceCharge() {

14 *return* areaServiceCharge;

15 }

16

17 *public* *void* setAreaServiceCharge(Map<String, Float> areaServiceCharge) {

18 *this*.areaServiceCharge = areaServiceCharge;

19 }

20

21

22 }

23

#### DayOutTravelService/src/main/java/com/spring/model/Travel.java

1 *package* com.spring.model;

2 *import* org.springframework.stereotype.Component;

3 *import* org.springframework.beans.factory.annotation.Autowired;

4 *import* org.springframework.beans.factory.annotation.Value;

5 *import* org.springframework.context.annotation.\*;

6

7 @Component

8

9 @PropertySource("classpath:charges.properties")

10

11 *public* *class* Travel {

12

13 *private* String tripId;

14 *private* *int* distanceTravelled;

15

16 @Value("${chargePerKm}")

17 *private* *float* chargePerKm;

18

19 @Autowired

20 *private* AdditionalChargeInfo additionalCharge;

21

22 *public* String getTripId() {

23 *return* tripId;

24 }

25 *public* *void* setTripId(String tripId) {

26 *this*.tripId = tripId;

27 }

28 *public* *int* getDistanceTravelled() {

29 *return* distanceTravelled;

30 }

31 *public* *void* setDistanceTravelled(*int* distanceTravelled) {

32 *this*.distanceTravelled = distanceTravelled;

33 }

34 *public* *float* getChargePerKm() {

35 *return* chargePerKm;

36 }

37 *public* *void* setChargePerKm(*float* chargePerKm) {

38 *this*.chargePerKm = chargePerKm;

39 }

40 *public* AdditionalChargeInfo getAdditionalCharge() {

41 *return* additionalCharge;

42 }

43 *public* *void* setAdditionalCharge(AdditionalChargeInfo additionalCharge) {

44 *this*.additionalCharge = additionalCharge;

45 }

46

47 }

48

#### DayOutTravelService/src/main/java/com/spring/service/TravelService.java

1 *package* com.spring.service;

2

3 *import* org.springframework.beans.factory.annotation.Autowired;

4 *import* org.springframework.context.ApplicationContext;

5 *import* org.springframework.context.annotation.AnnotationConfigApplicationContext;

6 *import* org.springframework.stereotype.Component;

7

8 *import* com.spring.bo.TravelBO;

9 *import* com.spring.config.ApplicationConfig;

10 *import* com.spring.exception.InvalidDistanceTravelledException;

11 *import* com.spring.model.Travel;

12

13 @Component

14 *public* *class* TravelService {

15

16 @Autowired

17 *private* TravelBO travelBoObj;

18

19 *public* TravelBO getTravelBoObj() {

20 *return* travelBoObj;

21 }

22

23 *public* *void* setTravelBoObj(TravelBO travelBoObj) {

24 *this*.travelBoObj = travelBoObj;

25 }

26

27 *public* *double* calculateTripCharge(String tripId,*int* distanceTravelled,String hraType) throws InvalidDistanceTravelledException {

28

29 *double* tripCharge=0.0;

30

31 *if*(distanceTravelled>0 && distanceTravelled<400){

32 ApplicationContext context = *new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

33

34 Travel t = context.getBean(Travel.*class*);

35 t.setChargePerKm(distanceTravelled);

36 t.setDistanceTravelled(distanceTravelled);

37

38 tripCharge = travelBoObj.calculateTripCharge(t, hraType);

39 }

40 *else*{

41 *throw* *new* InvalidDistanceTravelledException("Distance Travelled is not valid");

42 }

43 *return* tripCharge;

44 }

45

46 }

47

#### DayOutTravelService/src/main/resources/charges.properties

1 chargePerKm =10.0

## Grade

Reviewed on Tuesday, 27 April 2021, 3:11 AM by Automatic grade  
**Grade** 97.5 / 100  
**Assessment report**  
Fail 1 --FunctionalTest::Could not load ApplicationConfig. Check whether you have correctly configured  
[[+]](javascript:void(0);)**Grading and Feedback**