Book A Resort- Tested

**Grade settings**: Maximum grade: 100  
**Based on**: [Spring Core METADATA](https://cognizant.tekstac.com/mod/vpl/view.php?id=20396)  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes **Maximum execution time**: 120 s **Maximum memory used**: 1.50 GiB **Maximum execution file size**: 128 MiB

**Hill View Resort - Book a Room**

[**Click here to download the Code Skeleton**](https://cognizant.tekstac.com/pluginfile.php/68829/mod_vpl/intro/Book%20A%20Resort.zip)

Hill View Resort is a popular resort in Munnar.  During season, they have lots of customers requesting for Booking the resort. Help them by developing an application for maintaining their Customer and their booking details.

**Service 1:** Calculate Room rent

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

|  |  |
| --- | --- |
| bookingId | String |
| bookingDate | String |
| customer | Customer |
| roomType | String |
| noOfDays | int |

**Getters and setters** 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.

A one argument constructor is provided as part of the code skeleton. The **Customer** object should be autowired above the constructor in the **BookRoom** class via annotations.

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

|  |  |
| --- | --- |
| customerId | int |
| customerName | String |
| emailId | String |

**Getters and setters** 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.

**TariffInfo** class with the below private attributesis provided as a part of code skeleton

|  |  |
| --- | --- |
| tariffInfo | Map<String,Double> |

**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 tariffInfo  Map is available in **roomTariff.properties**file. Fetch the value from the property file and assign it to the Map tariffInfo in **TariffInfo**class. The values for this Map should be injected through**@Value annotation**. Do not change the value of **roomTypeWithTariff.map** in the property file.

|  |  |
| --- | --- |
| **Key – room type(String)** | **Value – Room Rent Per Day (Integer)** |
| Mini Suite | 5500 |
| Suite | 10000 |
| President Suite | 15000 |
| Villa | 25000 |

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

**Overview of Service 1:**

Write a method **public double calculateRoomRent(BookRoom** **bookRoomObj)** in**BookRoomBO class**which will return the**total room rent**for the noOfDays of stay. , this method should get the room rent (tariff) based on the roomType, which is available in the map, and calculate the room rent for the number of days of stay and return the same**.**

**For Example:**

If the **roomType**is **Villa**and**noOfDays**is**3 then**,

**Total room rent**= roomRent \* noOfDays;

**Total room rent  =**25000 \* 3 ==> 75000.0

**Assumption:** The value provided for roomType should be only the key values which are available in the map.

Create a class called **Driver** with the main method and get the inputs for BookRoom  and Customer detailsfrom the user. Get the object of **BookRoomService** class by loading **ApplicationConfig** class and invoke themethod**calculateRoomRent** (String bookingId,String bookingDate,int customerId,String customerName,String emailId,String roomType,int noOfDays) which is in the **BookRoomService** class to perform the calculation. Display the totalRoomRent which is returned from **calculateRoomRent** (bookRoomObj) method in**BookRoomBO class.**

**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| BookRoomService | double calculateRoomRent | String bookingId, String bookingDate, int customerId,  String customerName, String emailId, String roomType,    int noOfDays | double - totalRoomRent | InvalidRoomTypeException  **This Exception to be caught and thrown back to Main class** |
| BookRoomBO | double calculateRoomRent | BookRoom bookRoomObj | double - totalRoomRent |  |
| ApplicationConfig | Contains all the configurations related to Service |  |  |  |

**Business Rules & Validations:**

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

**private BookRoomBO bookRoomBOObj;**

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

In this **BookRoomService class**, the method**public double  calculateRoomRent(bookingId, bookingDate, customerId,  customerName, emailId, roomType, noOfDays)**accepts the bookingId, bookingDate, customerId,  customerName, emailId, roomType, noOfDays  as arguments .Validate the **roomType,[** **roomType values should be same as in the Map table]** if the roomType is valid get the BookRoom object and set the values for bookingId, bookingDate, customerId,  customerName, emailId, roomType, noOfDays  in that object. In case the roomTypeis not valid, a user-defined Exception InvalidRoomTypeException should be thrown with the message “**Room type is not valid**”.

If the validation is done call the method **calculateRoomRent (bookRoomObj) in BookRoomBO class** and perform the calculation.

**Limitations and Constraints:**

1.    **BookRoom , Customer and TariffInfo class**should be in**com.spring.model**package.

2.    **roomTariff.properties** file will be provided with the value of room type.  Fetch and assign values for **tariffInfo map**in **TariffInfo**class via **@Value annotation.**.

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

4.    **InvalidRoomTypeException** class should be in **com.spring.exception** package.

5.    **BookRoomService**class should be in**com.spring.service**package**.**

6.    **BookRoomBO class**should be in**com.spring.bo**package.

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

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

9.    **Customer** should be injected into **BookRoom**classvia constructor based Injection using appropriate annotation.

10. **BookRoomBO** should be injected into **BookRoomService** class via constructor based Injection using annotation.

**Sample Input Output 1:**

Enter the Booking Details

Booking ID

**B21**

Booking Date

**04/03/2021**

Customer ID

**192**

Customer Name

**Raghu**

Email ID

**raghu@gmail.com**

Room Type

**Mini Suite**

Number of days of stay

**6**

**Welcome Raghu Your choice is Mini Suite for 6 days You need to pay Rs. 33000.0**

**Sample Input Output 2:**

Enter the Booking Details

Booking ID

**B11**

Booking Date

**06/03/2021**

Customer ID

**12**

Customer Name

**Ashwin**

Email ID

**ashwin@gmail.com**

Room Type

**Single Room**

Number of days of stay

**4**

Room type is not valid

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

**Proposed grade: 69.25 / 100**  
**Result Description**  
Fail 1 --test11CalculateRoomRentInBookRoomBOForMiniSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Mini Suite  
Fail 2 --test12CalculateRoomRentInBookRoomBOForSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Suite  
Fail 3 --test13CalculateRoomRentInBookRoomBOForPresidentSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is President Suite  
Fail 4 --test14CalculateRoomRentInBookRoomBOForVilla::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Villa  
Fail 5 --test16CalculateRoomRentInBookRoomServiceForSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Suite  
Fail 6 --test17CalculateRoomRentInBookRoomServiceForPresidentSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is President Suite  
Fail 7 --test18CalculateRoomRentInBookRoomServiceForVilla::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Villa  
[[+]](javascript:void(0);)**SOURCE CODE ANALYZER REPORT**  
[[+]](javascript:void(0);)**Grading and Feedback**

#### **BookAResort/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>BookAResort</groupId>

4 <artifactId>BookAResort</artifactId>

5 <version>0.0.1-SNAPSHOT</version>

6 <name>BookAResort</name>

7

8 <dependencies>

9 <dependency>

10 <groupId>org.springframework</groupId>

11 <artifactId>spring-context</artifactId>

12 <version>4.3.10.RELEASE</version>

13 </dependency>

14

15 <dependency>

16 <groupId>junit</groupId>

17 <artifactId>junit</artifactId>

18 <version>4.12</version>

19 <scope>test</scope>

20 </dependency>

21 <dependency>

22 <groupId>xmlunit</groupId>

23 <artifactId>xmlunit</artifactId>

24 <version>1.6</version>

25 <scope>test</scope>

26 </dependency>

27

28

29 <!-- https://mvnrepository.com/artifact/com.github.stefanbirkner/system-rules -->

30 <dependency>

31 <groupId>com.github.stefanbirkner</groupId>

32 <artifactId>system-rules</artifactId>

33 <version>1.16.0</version>

34 <scope>test</scope>

35 </dependency>

36

37 <dependency>

38 <groupId>com.sun.xml.bind</groupId>

39 <artifactId>jaxb-impl</artifactId>

40 <version>2.1.13</version>

41 </dependency>

42 <!-- Thanks for using https://jar-download.com -->

43

44 <!-- https://mvnrepository.com/artifact/javax.xml.bind/jaxb-api -->

45 <dependency>

46 <groupId>javax.xml.bind</groupId>

47 <artifactId>jaxb-api</artifactId>

48 <version>2.1</version>

49 </dependency>

50

51 </dependencies>

52

53 <build>

54 <plugins>

55 <plugin>

56 <groupId>org.apache.maven.plugins</groupId>

57 <artifactId>maven-surefire-plugin</artifactId>

58 <version>3.0.0-M1</version>

59 <configuration>

60 <testFailureIgnore>true</testFailureIgnore>

61 </configuration>

62

63

64 </plugin>

65

66

67

68 </plugins>

69 </build>

70

71 <properties>

72 <maven.compiler.source>1.8</maven.compiler.source>

73 <maven.compiler.target>1.8</maven.compiler.target>

74 </properties>

75

76 </project>

#### **BookAResort/src/main/java/com/spring/bo/BookRoomBO.java**

1 *package* com.spring.bo;

2

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

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

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

6

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

8 *import* com.spring.model.BookRoom;

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

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

11

12 //use appropriate annotation to make this class as component class

13 @Component

14 *public* *class* BookRoomBO {

15

16 *public* *double* calculateRoomRent(BookRoom bookRoomObj) throws InvalidRoomTypeException {

17 *double* totalRoomRent=0;

18 // fill the code

19

20 ApplicationContext ac= *new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

21 TariffInfo t = ac.getBean(TariffInfo.*class*);

22

23

24 //TariffInfo t=new TariffInfo();

25

26 *double* rent= t.getTariffInfo().get(bookRoomObj.getRoomType()); //System.out.println(rent);

27 totalRoomRent= rent \* (*double*)bookRoomObj.getNoOfDays();

28

29 *return* totalRoomRent;

30 }

31

32 }

33

#### **BookAResort/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.BookRoomBO;

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

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

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

12 *import* com.spring.service.BookRoomService;

13

14 //Use appropriate annotation

15 @Configuration

16 @ComponentScan

17 @PropertySource("classpath:roomTariff.properties")

18 *public* *class* ApplicationConfig {

19

20

21 @Bean

22 *public* BookRoomService book() {

23 *return* *new* BookRoomService(*new* BookRoomBO());

24 }

25

26 /\*

27 \* @Bean BookRoom bookRoom() { return new BookRoom(new Customer()); }

28 \*/

29

30 @Bean

31 *public* TariffInfo tariff() { *return* *new* TariffInfo(); }

32

33

34

35 }

36

#### **BookAResort/src/main/java/com/spring/exception/InvalidRoomTypeException.java**

1 *package* com.spring.exception;

2

3 *public* *class* InvalidRoomTypeException *extends* Exception {

4

5 *public* InvalidRoomTypeException(String msg) {

6

7 // fill the code

8 *super*(msg);

9 }

10

11 }

12

#### **BookAResort/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.InvalidRoomTypeException;

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

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

12 *import* com.spring.service.BookRoomService;

13

14 *public* *class* Driver {

15

16 *public* *static* *void* main(String[] args) throws InvalidRoomTypeException {

17

18 // fill the code

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

20 ApplicationContext ac= *new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

21

22 Customer cr= *new* Customer();

23

24 System.out.println("Enter the Booking Details");

25 System.out.println("Booking ID");

26 String bid= sc.nextLine(); //System.out.println(bid);

27

28 System.out.println("Booking Date");

29 String bdate= sc.nextLine(); //System.out.println(bdate);

30

31 System.out.println("Customer ID");

32 *int* cid= Integer.parseInt(sc.nextLine().trim());//System.out.println(cid);

33

34 //sc.next();

35 System.out.println("Customer Name");

36 String cname= sc.nextLine(); //System.out.println(cname);

37

38 System.out.println("Email ID");

39 String email=sc.nextLine(); //System.out.println(email);

40 cr.setCustomerId(cid);

41 cr.setCustomerName(cname);

42 cr.setEmailId(email);

43

44

45

46 System.out.println("Room Type");

47 String rtype=sc.nextLine(); //System.out.println(rtype);

48

49 System.out.println("Number of days of stay");

50 //sc.next();

51 *int* no =Integer.parseInt(sc.nextLine().trim()); //System.out.println(no);

52

53

54

55

56 BookRoom br= *new* BookRoom(cr);

57 br.setBookingDate(bdate);

58 br.setBookingId(bid);

59 br.setNoOfDays(no);

60 br.setRoomType(rtype);

61

62 BookRoomService book=(BookRoomService) ac.getBean(BookRoomService.*class*);

63 *double* x=book.calculateRoomRent(bid, bdate, cid, cname, email, rtype, no);

64

65 System.out.println("Welcome "+cr.getCustomerName()+" Your choice is "+br.getRoomType()+" for "+br.getNoOfDays()+

66 " days You need to pay Rs. "+x);

67

68 }

69

70 }

71

#### **BookAResort/src/main/java/com/spring/model/BookRoom.java**

1 *package* com.spring.model;

2

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

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

5

6 //pojo class with required attributes,getters and setters

7 //use appropriate annotation to make this class as component class

8 @Component

9 *public* *class* BookRoom {

10

11 *private* String bookingId;

12 *private* String bookingDate;

13 *private* Customer customer;

14 *private* String roomType;

15 *private* *int* noOfDays;

16

17

18 // fill the code

19 @Autowired

20 *public* BookRoom(Customer customer) {

21 *this*.customer=customer;

22 }

23 *public* String getBookingId() {

24 *return* bookingId;

25 }

26 *public* *void* setBookingId(String bookingId) {

27 *this*.bookingId = bookingId;

28 }

29 *public* String getBookingDate() {

30 *return* bookingDate;

31 }

32 *public* *void* setBookingDate(String bookingDate) {

33 *this*.bookingDate = bookingDate;

34 }

35 *public* Customer getCustomer() {

36 *return* customer;

37 }

38 *public* *void* setCustomer(Customer customer) {

39 *this*.customer = customer;

40 }

41 *public* String getRoomType() {

42 *return* roomType;

43 }

44 *public* *void* setRoomType(String roomType) {

45 *this*.roomType = roomType;

46 }

47 *public* *int* getNoOfDays() {

48 *return* noOfDays;

49 }

50 *public* *void* setNoOfDays(*int* noOfDays) {

51 *this*.noOfDays = noOfDays;

52 }

53

54

55 }

56

57

#### **BookAResort/src/main/java/com/spring/model/Customer.java**

1 *package* com.spring.model;

2

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

4

5 //pojo class with required attributes,getters and setters

6 //use appropriate annotation to make this class as component class

7 @Component

8 *public* *class* Customer {

9

10 *private* *int* customerId;

11 *private* String customerName;

12 *private* String emailId;

13

14 *public* *int* getCustomerId() {

15 *return* customerId;

16 }

17 *public* *void* setCustomerId(*int* customerId) {

18 *this*.customerId = customerId;

19 }

20 *public* String getCustomerName() {

21 *return* customerName;

22 }

23 *public* *void* setCustomerName(String customerName) {

24 *this*.customerName = customerName;

25 }

26

27 *public* String getEmailId() {

28 *return* emailId;

29 }

30 *public* *void* setEmailId(String emailId) {

31 *this*.emailId = emailId;

32 };

33

34

35 }

36

#### **BookAResort/src/main/java/com/spring/model/TariffInfo.java**

1 *package* com.spring.model;

2

3 *import* java.util.Map;

4

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

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

7

8 //pojo class with required attributes,getters and setters

9 //use appropriate annotation to make this class as component class

10 @Component

11 *public* *class* TariffInfo {

12

13 //Fill the code

14 @Value("#{${roomTypeWithTariff.map}}") //"#{${...}}"

15 *private* Map<String,Double> tariffInfo;

16

17 *public* Map<String, Double> getTariffInfo() {

18 *return* tariffInfo;

19 }

20

21 *public* *void* setTariffInfo(Map<String, Double> tariffInfo) {

22 *this*.tariffInfo = tariffInfo;

23 }

24

25 }

26

#### **BookAResort/src/main/java/com/spring/service/BookRoomService.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.context.annotation.Bean;

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

8

9 *import* com.spring.bo.BookRoomBO;

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

11 *import* com.spring.exception.InvalidRoomTypeException;

12 *import* com.spring.model.BookRoom;

13 *import* com.spring.model.Customer;

14 *import* com.spring.model.TariffInfo;

15

16 //use appropriate annotation to make this class as component class

17 @Component

18 *public* *class* BookRoomService {

19

20 *private* BookRoomBO bookRoomBOObj;

21

22 //fill the code

23 @Autowired

24 *public* BookRoomService(BookRoomBO bookRoomBOObj) {

25 *super*();

26 *this*.bookRoomBOObj = bookRoomBOObj;

27 }

28

29

30 *public* BookRoomBO getBookRoomBOObj() {

31 *return* bookRoomBOObj;

32 }

33

34

35 *public* *void* setBookRoomBOObj(BookRoomBO bookRoomBOObj) {

36 *this*.bookRoomBOObj = bookRoomBOObj;

37 }

38

39

40 *public* *double* calculateRoomRent(String bookingId,String bookingDate,*int* customerId,String customerName,String emailId,String roomType,*int* noOfDays)

41 throws InvalidRoomTypeException

42 {

43

44 *double* totalRoomRent=0;

45

46 // fill the code

47 ApplicationContext ac= *new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

48 TariffInfo t = ac.getBean(TariffInfo.*class*);

49

50 //TariffInfo t=new TariffInfo();

51 //BookRoomBO bo=new BookRoomBO();

52

53 //System.out.println(t.getTariffInfo());

54

55 *if*(t.getTariffInfo().containsKey(roomType)) {

56 Customer c=*new* Customer();

57 c.setCustomerId(customerId);

58 c.setCustomerName(customerName);

59 c.setEmailId(emailId);

60

61 BookRoom br= *new* BookRoom(c);

62

63 br.setBookingId(bookingId);

64 br.setBookingDate(bookingDate);

65 br.setRoomType(roomType);

66 br.setNoOfDays(noOfDays);

67

68 totalRoomRent= bookRoomBOObj.calculateRoomRent(br);

69

70 }

71 *else*

72 *throw* *new* InvalidRoomTypeException("Room type is not valid");

73

74 *return* totalRoomRent;

75 }

76

77 }

78

#### **BookAResort/src/main/resources/roomTariff.properties**

1 roomTypeWithTariff.map={'Mini Suite':'5500', 'Suite':'10000', 'President Suite':'15000','Villa':25000}

## Grade

Reviewed on Friday, 21 May 2021, 6:33 PM by Automatic grade  
**Grade** 69.25 / 100  
**Assessment report**  
Fail 1 --test11CalculateRoomRentInBookRoomBOForMiniSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Mini Suite  
Fail 2 --test12CalculateRoomRentInBookRoomBOForSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Suite  
Fail 3 --test13CalculateRoomRentInBookRoomBOForPresidentSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is President Suite  
Fail 4 --test14CalculateRoomRentInBookRoomBOForVilla::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Villa  
Fail 5 --test16CalculateRoomRentInBookRoomServiceForSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Suite  
Fail 6 --test17CalculateRoomRentInBookRoomServiceForPresidentSuite::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is President Suite  
Fail 7 --test18CalculateRoomRentInBookRoomServiceForVilla::Check the logic of calculateRoomRent method in BookRoomBO class when roomType is Villa  
[[+]](javascript:void(0);)**SOURCE CODE ANALYZER REPORT**  
[[+]](javascript:void(0);)**Grading and Feedback**