FurnTurn Outlet

**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

**Furn Turn Outlet – Calculate total bill for the furniture**

[**Click here to download the Code Skeleton**](https://cognizant.tekstac.com/pluginfile.php/19293/mod_vpl/intro/FurnTurnOutlet.zip)

FurnTurn is one of the popular furniture outlets in the city. They are in need of an application to calculate the bill amount to be paid by the customer based on the type of wood. Each type of wood has different discount percentage. You being their software consultant have been approached by them to develop an application which can be used for managing their business.

**Service 1:** Calculating bill amount

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

|  |  |
| --- | --- |
| productId | String |
| productName | String |
| mrpValue | double |
| dimension | String |
| shopDetails | Shop |

**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.  One argument constructor is provided as the part the code skeleton. The **Shop** object should be autowired above the constructor in **Product** class via annotations.

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

|  |  |
| --- | --- |
| shopName | String |
| shopLocation | String |
| discountDetails | Map<String,Integer> |

**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 attribute should be configured using @Value annotation with the below **values**

|  |  |
| --- | --- |
| ShopName | FurnTurn Bang |
| shopLocation | Bangalore |

The Map should be configured using @Value annotation with the below **key-values**

|  |  |
| --- | --- |
| **Key – Wood Type(String)** | **Value – Discount Percentage (Integer)** |
| Teak | 3 |
| Sheesham | 5 |
| RubberWood | 2 |
| Wenge | 8 |
| Pine | 4 |

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 calculateBill(Product** **furObj, String woodType)** in**ProductBO class**which will return the**bill amount**, this method should get the discount rate based on the woodType which is available in the map and calculate the discount amount based on mrp. Reduce the discount amount from mrp to calculate the bill amountand return the same**.**

**For Example:**

If the **wood type** is **Sheesham**and**mrp**is**30000 then**,

billAmount = mrp - (mrp \* 5) / 100.0;

**billAmount= 30000.0 - (30000.0 \* 5) /100.0 ==> 28500.0**

**Assumption:** Wood type 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 like **productId**, **productName**, **mrpValue, dimension** and **woodType** from the user. Get the **ProductService** class object by loading **ApplicationConfig** class and invoke themethod**calculateBill(productId, productName, mrpValue, dimension**,**woodType)** which is in the **ProductService** class to perform the implementation. Display the total **bill** **amount** which is returned from **calculateBill(Product obj,String woodType)**method in**ProductBO class.**

**Technical Specifications:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Method Name** | **Input** | **Output** | **Exception** |
| ProductService | double calculateBill | String productId, String productName, double mrpValue,  String dimension, String woodType | double - billAmount | InvalidProductIdException  **This Exception to be caught and thrown back to Main class** |
| ProductBO | double calculateBill | Product furObj, String woodType | double - billAmount |  |
| ApplicationConfig | Contains all the configurations related to Service |  |  |  |

**Business Rules & Validations:**

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

**private ProductBO productBOObj;**

**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 **ProductBO** object should be **autowired** above the **constructor** via **annotations**.

In this **ProductService class**, the method**public double calculateBill(productId, productName, mrpValue, dimension,woodType)**accepts productId, productName, mrpValue, dimension and woodType as the arguments .Validate the productId**,[productId should be of length 8 and should be only digits]** if the **productId** is **valid** get the Furniture object and set the productId, productName, mrpValue and dimension  in that object. In case the **productId** is NOT **valid**, a user-defined Exception **InvalidProductIdException**should be thrown with message "**Invalid Product ID"**.

If the validation is done call the method **calculateBill (productObj, woodType) in ProductBO class** and perform the implementation.

**Limitations and Constraints:**

1.    **Product and Shop class**should be in**com.spring.model**package.

2.    The **discount details** for each wood type should be declared as a MAP with the given value and should be injected using @Value annotation above the attribute.

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

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

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

6.    **ProductBO 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.    **Shop** should be injected into **Product**classvia constructor based Injection using appropriate annotation.

10.  **ProductBO** should be injected into **ProductService** class via constructor based Injection using annotation.

**Sample Input Output 1:**

Enter the product id:

**51236524**

Enter the product name:

**Dinning table**

Enter the mrp value:

**30000**

Enter the dimension details:

**4X3**

Enter the wood type:

**Sheesham // wood type available in Map which is case sensitive**

Amount to be paid is: 28500.0

**Sample Input Output 2:**

Enter the product id:

**512365AB**

Enter the  product name:

**Dinning table**

Enter the mrp value:

**30000**

Enter the dimension details:

**4X3**

Enter the wood type:

**Sheesham // wood type available in Map which is case sensitive**

Invalid Product ID

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

**Proposed grade: 100.0 / 100**  
**Result Description**  
[[+]](javascript:void(0);)**Grading and Feedback**

#### FurnTurnOutlet/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>FurnTurnOutlet</groupId>

4 <artifactId>FurnTurnOutlet</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>

#### FurnTurnOutlet/src/main/java/com/spring/bo/ProductBO.java

1 *package* com.spring.bo;

2

3 *import* java.util.Map;

4

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

6

7 *import* com.spring.model.Product;

8

9 @Component

10 *public* *class* ProductBO {

11

12 *public* *double* calculateBill(Product furObj,String woodType) {

13

14 *double* amount=0;

15 Map<String,Integer> map=furObj.getShopDetails().getDiscountDetails();

16 *if*(map.containsKey(woodType)) {

17 *int* d=map.get(woodType);

18 amount=furObj.getMrpValue()-(furObj.getMrpValue()\*d)/100;

19 }

20 // fill the code

21

22 *return* amount;

23 }

24

25 }

26

#### FurnTurnOutlet/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

7 *import* com.spring.bo.ProductBO;

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

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

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

11

12 // fill the code

13 @Configuration

14 @ComponentScan

15 *public* *class* ApplicationConfig {

16 @Bean

17 *public* Shop shop() {

18 *return* *new* Shop();

19 }

20

21 @Bean

22 *public* Product product() {

23 *return* *new* Product(shop());

24 }

25

26 @Bean

27 *public* ProductBO productBO() {

28 *return* *new* ProductBO();

29 }

30

31 @Bean

32 *public* ProductService productService() {

33 *return* *new* ProductService(productBO());

34 }

35 }

36

#### FurnTurnOutlet/src/main/java/com/spring/exception/InvalidProductIdException.java

1 *package* com.spring.exception;

2

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

4

5 @Component

6 *public* *class* InvalidProductIdException *extends* Exception {

7

8 *public* InvalidProductIdException(String msg) {

9 *super*(msg);

10

11 // fill the code

12 }

13

14 }

15

#### FurnTurnOutlet/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.ConfigurableApplicationContext;

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

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

8

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

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

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

12

13 @Component

14 *public* *class* Driver {

15

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

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

18 System.out.println("Enter the product id:");

19 String productId=sc.nextLine();

20 System.out.println("Enter the product name:");

21 String productName=sc.nextLine();

22 System.out.println("Enter the mrp value:");

23 *double* mrp=sc.nextDouble();

24 sc.nextLine();

25 System.out.println("Enter the dimension details:");

26 String dimension=sc.nextLine();

27 System.out.println("Enter the wood type:");

28 String woodType=sc.nextLine();

29 ConfigurableApplicationContext context=*new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

30 ProductService product=(ProductService) context.getBean(ProductService.*class*);

31 *try* {

32 *double* d=product.calculateBill(productId,productName,mrp,dimension,woodType);

33 System.out.println("Amount to be paid is:"+d);

34

35 }*catch*(InvalidProductIdException e) {

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

37 }

38

39

40 // fill the code

41

42

43 }

44

45 }

46

#### FurnTurnOutlet/src/main/java/com/spring/model/Product.java

1 *package* com.spring.model;

2

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

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

5

6 @Component

7 *public* *class* Product {

8

9 *protected* String productId;

10 *protected* String productName;

11 *protected* *double* mrpValue;

12 *protected* Shop shopDetails;

13 *private* String dimension;

14

15 @Autowired

16 *public* Product(Shop shopDetails) {

17 *super*();

18 *this*.shopDetails = shopDetails;

19 }

20 *public* String getDimension() {

21 *return* dimension;

22 }

23 *public* *void* setDimension(String dimension) {

24 *this*.dimension = dimension;

25 }

26 *public* String getProductId() {

27 *return* productId;

28 }

29 *public* *void* setProductId(String productId) {

30 *this*.productId = productId;

31 }

32 *public* String getProductName() {

33 *return* productName;

34 }

35 *public* *void* setProductName(String productName) {

36 *this*.productName = productName;

37 }

38 *public* *double* getMrpValue() {

39 *return* mrpValue;

40 }

41 *public* *void* setMrpValue(*double* mrpValue) {

42 *this*.mrpValue = mrpValue;

43 }

44 *public* Shop getShopDetails() {

45 *return* shopDetails;

46 }

47 *public* *void* setShopDetails(Shop shopDetails) {

48 *this*.shopDetails = shopDetails;

49 }

50

51 }

52

53

#### FurnTurnOutlet/src/main/java/com/spring/model/Shop.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 @Component

9 *public* *class* Shop {

10

11 @Value("FurnTurn Bang")

12 *private* String shopName;

13

14 @Value("Bangalore")

15 *private* String shopLocation;

16

17 @Value("#{${discountDetails:{Teak:'3',"

18 + "Sheesham:'5',"

19 + "RubberWood:'2',"

20 + "Wenge:'8',"

21 + "Pine:'4'}}}")

22 *private* Map<String,Integer> discountDetails;

23

24 *public* String getShopName() {

25 *return* shopName;

26 }

27 *public* *void* setShopName(String shopName) {

28 *this*.shopName = shopName;

29 }

30 *public* String getShopLocation() {

31 *return* shopLocation;

32 }

33 *public* *void* setShopLocation(String shopLocation) {

34 *this*.shopLocation = shopLocation;

35 }

36 *public* Map<String, Integer> getDiscountDetails() {

37 *return* discountDetails;

38 }

39 *public* *void* setDiscountDetails(Map<String, Integer> discountDetails) {

40 *this*.discountDetails = discountDetails;

41 }

42

43 }

44

#### FurnTurnOutlet/src/main/java/com/spring/service/ProductService.java

1 *package* com.spring.service;

2

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

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

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

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

7 *import* com.spring.bo.ProductBO;

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

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

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

11

12

13 @Component

14 *public* *class* ProductService{

15

16 *private* ProductBO productBOObj;

17

18 @Autowired

19 *public* ProductService(ProductBO productBOObj) {

20 *super*();

21 *this*.productBOObj = productBOObj;

22 }

23

24 *public* ProductBO getProductBOObj() {

25 *return* productBOObj;

26 }

27

28 *public* *void* setProductBOObj(ProductBO productBOObj) {

29 *this*.productBOObj = productBOObj;

30 }

31

32 *public* *double* calculateBill(String productId,String productName,*double* mrpValue,String dimension, String woodType) throws InvalidProductIdException {

33

34 *double* amount=0;

35

36 *if*(productId.matches("[0-9]+")) {

37 ConfigurableApplicationContext context=*new* AnnotationConfigApplicationContext(ApplicationConfig.*class*);

38 Product product=(Product)context.getBean(Product.*class*);

39 product.setDimension(dimension);

40 product.setMrpValue(mrpValue);

41 product.setProductName(productName);

42 product.setProductId(productId);

43 amount=productBOObj.calculateBill(product, woodType);

44 }

45 *else* {

46 *throw* *new* InvalidProductIdException("Invalid Product ID");

47 }

48 // fill the code

49 *return* amount;

50 }

51

52 }

## Grade

Reviewed on Wednesday, 24 March 2021, 5:56 PM by Automatic grade  
**Grade** 100 / 100  
**Assessment report**  
[[+]](javascript:void(0);)**Grading and Feedback**