Package BuildingCentreReceipt

Class BuildingCentreProductBase

java.lang.Object
    BuildingCentreReceipt.BuildingCentreProductBase

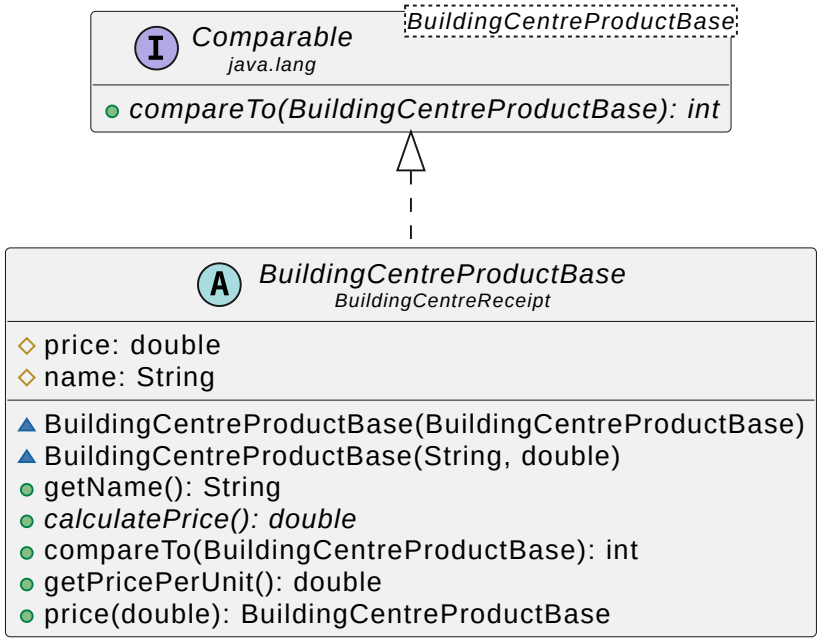
All Implemented Interfaces:

`java.lang.Comparable<BuildingCentreProductBase>`

Direct Known Subclasses:

`CraftsmanSupport`, `Material`, `Tool`

```
public abstract class
BuildingCentreProductBase
extends java.lang.Object
implements
```



UMLDoclet 2.1.0, PlantUML 1.2022.14

`java.lang.Comparable<BuildingCentreProductBase>`

A product in the building centre. A product is anything that is offered for sale including physical items and services.
This class is the base class of all products in the building centre.
Naturally, products are sorted by (total) price.

Field Summary

Fields

Modifier and Type	Field	Description

Constructor Summary

Constructors

Constructor	Description
BuildingCentreProductBase (**BuildingCentreProductBase** orig)	copy constructor
BuildingCentreProductBase (java.lang.String name, double pricePerUnit)	Constructs a product with a specified name and price per unit In case of illegal arguments an exception is thrown.

Method Summary

All Methods | **Instance Methods** | **Abstract Methods** | **Concrete Methods**

Modifier and Type	Method	Description
abstract double	**calculatePrice**()	calculates the price of this product.
int	**compareTo** (**BuildingCentreProductBase** o)	Compares this product to another product by (total) price.
java.lang.String	**getName**()	Get the name of this product.
double	**getPricePerUnit**()	get the price per unit of this product
BuildingCentreProductBase	**price**(double price)	sets the price of this product. Price is set if and only if it is apositive value.
java.lang.String	**toString**()	String representation of this product. The string

Methods declared in class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

price

`protected double price`

the price per unit of this product, defaults to 1

name

`protected java.lang.String name`

the name of this product

Constructor Detail

BuildingCentreProductBase

`BuildingCentreProductBase(BuildingCentreProductBase orig)`

copy constructor

Parameters:

`orig` - the product to copy

Constructs a product with a specified name and price per unit
In case of illegal arguments an exception is thrown.

Parameters:

`name` - the name of this product

`pricePerUnit` - the price per unit of this product in Euro (EUR)

Throws:

`java.lang.IllegalArgumentException` - if the price per unit is less than one

Method Detail

getName

`public java.lang.String getName()`

Get the name of this product.

Returns:

the name

calculatePrice

`public abstract double calculatePrice()`

calculates the price of this product. The actual calculation is performed in concrete subclasses.

Returns:

the price of this product in euro (EUR)

toString

`public java.lang.String toString()`

compareTo

```
public int compareTo(BuildingCentreProductBase o)
```

Compares this product to another product by (total) price.

Specified by:

`compareTo` in interface `java.lang.Comparable<BuildingCentreProductBase>`

Parameters:

`o` - the product to compare this procuct to

Returns:

the "distance" in terms of price betwee this product and the other product

getPricePerUnit

```
public double getPricePerUnit()
```

get the price per unit of this product

Returns:

the price per unit

price

```
public BuildingCentreProductBase price(double price)
```

sets the price of this product.
Price is set if and only if it is apositive value. Illegal arguments are ignored.
This method is intended for method chaining.

Parameters:

`price` - the price to set

Returns:

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD