

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Interface LoadTask.VendingMachineLoaderCallback

All Known Implementing Classes:

[VendingMachineConsole](#)

Enclosing class:

[LoadTask](#)

public static interface LoadTask.VendingMachineLoaderCallback
Callback interface for load completion.

• Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type

Method

Description

void

onLoadComplete (VendingMachine machine)

Called when the load is complete.

• Method Details

◆ onLoadComplete

void onLoadComplete(VendingMachine machine)

Called when the load is complete.

Parameters:

machine - the loaded vending machine

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Class LoadTask

[java.lang.Object](#)

LoadTask

All Implemented Interfaces:

[Runnable](#)

public class LoadTask extends [Object](#) implements [Runnable](#)
A task that loads the state of a vending machine from a file.

• Nested Class Summary

Nested Classes

Modifier and Type

Class

Description

static interface

[LoadTask.VendingMachineLoaderCallback](#)

Callback interface for load completion.

• Constructor Summary

Constructors

Constructor

Description

[LoadTask](#)([String](#) filePath,

[LoadTask.VendingMachineLoaderCallback](#) callback)

Constructs a new LoadTask.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[run](#)()

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

• Constructor Details

◆ LoadTask

public LoadTask([String](#) filePath, [LoadTask.VendingMachineLoaderCallback](#) callback)

Constructs a new LoadTask.

Parameters:

filePath - the path of the file to load the state from

callback - the callback to invoke when the load is complete

• Method Details

◆ **run**

public void run()

Specified by:

run in interface Runnable

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Class SaveTask

[java.lang.Object](#)

SaveTask

All Implemented Interfaces:

[Runnable](#)

public class SaveTask extends [Object](#) implements [Runnable](#)
A task that saves the state of a vending machine to a file.

• Constructor Summary

Constructors

Constructor

Description

SaveTask(String filePath, VendingMachine machine)

Constructs a new SaveTask.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

run()

Methods inherited from class java.lang.Object

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

• Constructor Details

◆ SaveTask

public SaveTask(String filePath, VendingMachine machine)

Constructs a new SaveTask.

Parameters:

filePath - the path of the file to save the state to

machine - the vending machine to save

• Method Details

◆ run

public void run()

Specified by:

run in interface Runnable

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Class VendingMachineConsole

[java.lang.Object](#)

VendingMachineConsole

All Implemented Interfaces:

[LoadTask.VendingMachineLoaderCallback](#)

public class VendingMachineConsole extends [Object](#) implements [LoadTask.VendingMachineLoaderCallback](#)
Console application for managing vending machines.

• Constructor Summary

Constructors

Constructor

Description

[VendingMachineConsole\(\)](#)

• Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

static void

[main\(String\[\] args\)](#)

void

[onLoadComplete\(VendingMachine machine\)](#)

Called when the load is complete.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

• Constructor Details

♦ **VendingMachineConsole**

public VendingMachineConsole()

• Method Details

♦ **main**

public static void main([String\[\] args](#))

♦ **onLoadComplete**

public void onLoadComplete([VendingMachine machine](#))

Description copied from interface: [LoadTask.VendingMachineLoaderCallback](#)

Called when the load is complete.

Specified by:

[onLoadComplete](#) in

interface [LoadTask.VendingMachineLoaderCallback](#)

Parameters:

machine - the loaded vending machine

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

All Classes and Interfaces

All Classes and InterfacesInterfacesClassesException Classes

Class

Description

[CoffeeMachine](#)

A specialized vending machine that dispenses coffee.

[Displayable](#)

An interface that represents a displayable and refillable vending machine.

[InsufficientResourcesException](#)

This exception is thrown when there are insufficient resources to complete an operation.

[LoadTask](#)

A task that loads the state of a vending machine from a file.

[LoadTask.VendingMachineLoaderCallback](#)

Callback interface for load completion.

[ProductNameAlreadyExistsException](#)

This exception is thrown when a product name already exists in the vending machine.

[ProductNotFoundException](#)

This exception is thrown when a product is not found in the vending machine.

[Refillable](#)

An interface that represents a refillable vending machine.

[SandwichMachine](#)

A specialized vending machine that dispenses sandwiches.

[SaveTask](#)

A task that saves the state of a vending machine to a file.

[VendingMachine](#)

Represents a general vending machine.

[VendingMachineConsole](#)

Console application for managing vending machines.

[VendingMachineException](#)

Base class for exceptions thrown by the vending machine.

[VendingMachineFactory](#)

Factory class for creating vending machines.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

All Packages

Package Summary

Package

Description

[Unnamed Package](#)

[exceptions](#)

[interfaces](#)

[specializedmachines](#)

[Templates](#)

[vendingmachines](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- [Help:](#)
 - ◆ [Navigation](#)
 - ◆ [Pages](#)

- [Help:](#)
- [Navigation](#) |
- [Pages](#)

SEARCH

JavaDoc Help

- [Navigation:](#)
 - ◆ [Search](#)
- [Kinds of Pages:](#)
 - ◆ [Overview](#)
 - ◆ [Package](#)
 - ◆ [Class or Interface](#)
 - ◆ [Other Files](#)
 - ◆ [Tree \(Class Hierarchy\)](#)
 - ◆ [Serialized Form](#)
 - ◆ [All Packages](#)
 - ◆ [All Classes and Interfaces](#)
 - ◆ [Index](#)

Navigation

Starting from the [Overview](#) page, you can browse the documentation using the links in each page, and in the navigation bar at the top of each page. The [Index](#) and Search box allow you to navigate to specific declarations and summary pages, including: [All Packages](#), [All Classes and Interfaces](#)

Search

You can search for definitions of modules, packages, types, fields, methods, system properties and other terms defined in the API. These items can be searched using part or all of the name, optionally using "camelCase" abbreviations, or multiple search terms separated by whitespace. Some examples:

- "j.l.obj" matches "java.lang.Object"

- `"InpStr"` matches `"java.io.InputStream"`
- `"math exact long"` matches `"java.lang.Math.absExact(long)"`

Refer to the [Javadoc Search Specification](#) for a full description of search features.

Kinds of Pages

The following sections describe the different kinds of pages in this collection.

Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. These pages may contain the following categories:

- Interfaces
- Classes
- Enum Classes
- Exception Classes
- Annotation Interfaces

Class or Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a declaration and description, member summary tables, and detailed member descriptions. Entries in each of these sections are omitted if they are empty or not applicable.

- Class Inheritance Diagram
 - Direct Subclasses
 - All Known Subinterfaces
 - All Known Implementing Classes
 - Class or Interface Declaration
 - Class or Interface Description
-
- Nested Class Summary
 - Enum Constant Summary
 - Field Summary
 - Property Summary
 - Constructor Summary
 - Method Summary
 - Required Element Summary
 - Optional Element Summary

- Enum Constant Details
- Field Details
- Property Details
- Constructor Details
- Method Details
- Element Details

Note: Annotation interfaces have required and optional elements, but not methods. Only enum classes have enum constants. The components of a record class are displayed as part of the declaration of the record class. Properties are a feature of JavaFX.

The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Other Files

Packages and modules may contain pages with additional information related to the declarations nearby.

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. Classes are organized by inheritance structure starting with `java.lang.Object`. Interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on TREE displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking on TREE displays the hierarchy for only that package.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to those who implement rather than use the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See Also" section of the class description.

All Packages

The [All Packages](#) page contains an alphabetic index of all packages contained in the documentation.

All Classes and Interfaces

The [All Classes and Interfaces](#) page contains an alphabetic index of all classes and interfaces contained in the documentation, including annotation interfaces, enum classes, and record classes.

Index

The [Index](#) contains an alphabetic index of all classes, interfaces, constructors, methods, and fields in the documentation, as well as summary pages such as [All Packages](#), [All Classes and Interfaces](#).

LoadTask.VendingMachineLoaderCallback

This help file applies to API documentation generated by the standard doclet.

JavaScript is disabled on your browser.

Skip navigation links

- Overview
- Package
- Class
- Tree
- Index
- Help

SEARCH

Packages

Package

Description

Unnamed Package

exceptions

interfaces

specializedmachines

Templates

vendingmachines

JavaScript is disabled on your browser.

[index.html](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For All Packages

Package Hierarchies:

- [exceptions](#),
- [interfaces](#),
- [specializedmachines](#),
- [Templates](#),
- [vendingmachines](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [LoadTask](#) (implements [java.lang.Runnable](#))
 - ◆ [SaveTask](#) (implements [java.lang.Runnable](#))
 - ◆ [java.lang.Throwable](#) (implements [java.io.Serializable](#))
 - ◇ [java.lang.Exception](#)
 - [exceptions.VendingMachineException](#)
 - [exceptions.InsufficientResourcesException](#)
 - [exceptions.ProductNameAlreadyExistsException](#)
 - [exceptions.ProductNotFoundException](#)
 - ◆ [vendingmachines.VendingMachine](#) (implements [java.lang.Cloneable](#), [interfaces.Displayable](#), [java.io.Serializable](#))
 - ◇ [specializedmachines.CoffeeMachine](#)
 - ◇ [specializedmachines.SandwichMachine](#)
 - ◆ [VendingMachineConsole](#) (implements [LoadTask.VendingMachineLoaderCallback](#))
 - ◆ [Templates.VendingMachineFactory](#)

Interface Hierarchy

- [LoadTask.VendingMachineLoaderCallback](#)
- [interfaces.Refillable](#)
 - ◆ [interfaces.Displayable](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)

- Package:
- [Description](#)
- [Related Packages](#)
- [Classes and Interfaces](#)

SEARCH

Unnamed Package

- All Classes and InterfacesInterfacesClasses
 - Class
 - Description
 - [LoadTask](#)
A task that loads the state of a vending machine from a file.
 - [LoadTask.VendingMachineLoaderCallback](#)
Callback interface for load completion.
 - [SaveTask](#)
A task that saves the state of a vending machine to a file.
 - [VendingMachineConsole](#)
Console application for managing vending machines.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Unnamed Package

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [LoadTask](#) (implements [java.lang.Runnable](#))
 - ◆ [SaveTask](#) (implements [java.lang.Runnable](#))
 - ◆ [VendingMachineConsole](#) (implements [LoadTask.VendingMachineLoaderCallback](#))

Interface Hierarchy

- [LoadTask.VendingMachineLoaderCallback](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

Search

Additional resources

The [help page](#) provides an introduction to the scope and syntax of JavaDoc search.

You can use the <ctrl> or <cmd> keys in combination with the left and right arrow keys to switch between result tabs in this page.

The URL template below may be used to configure this page as a search engine in browsers that support this feature. It has been tested to work in Google Chrome and Mozilla Firefox. Note that other browsers may not support this feature or require a different URL format.

linkCopy

Redirect to first result

Loading search index...

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Serialized Form

• **Package [exceptions](#)**

♦ **Exception Class [exceptions.InsufficientResourcesException](#)**

class `InsufficientResourcesException` extends [VendingMachineException](#) implements [Serializable](#)

◊ **Serialized Fields**

. `operation`

`String operation`

♦ **Exception Class [exceptions.ProductNameAlreadyExistsException](#)**

class `ProductNameAlreadyExistsException` extends [VendingMachineException](#) implements [Serializable](#)

♦ **Exception Class [exceptions.ProductNotFoundException](#)**

class `ProductNotFoundException` extends [VendingMachineException](#) implements [Serializable](#)

♦ **Exception Class [exceptions.VendingMachineException](#)**

class `VendingMachineException` extends [Exception](#) implements [Serializable](#)

• **Package [specializedmachines](#)**

♦ **Class [specializedmachines.CoffeeMachine](#)**

class `CoffeeMachine` extends [VendingMachine](#) implements [Serializable](#)
serialVersionUID:

1L

◊ **Serialized Fields**

LoadTask.VendingMachineLoaderCallback

. beanLevel

int beanLevel

. sugarLevel

int sugarLevel

. waterLevel

int waterLevel

♦ Class specializedmachines.SandwichMachine

class SandwichMachine extends VendingMachine implements Serializable
serialVersionUID:

1L

♦ Serialized Fields

. breadQuantity

int breadQuantity

. cheeseQuantity

int cheeseQuantity

. hamQuantity

int hamQuantity

• Package vendingmachines

♦ Class vendingmachines.VendingMachine

class VendingMachine extends Object implements Serializable
serialVersionUID:

1L

♦ Serialized Fields

. productAmount

int productAmount

. products

Map<String, vendingmachines.VendingMachine.Product> products

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [vendingmachines](#)

Class VendingMachine

[java.lang.Object](#)

vendingmachines.VendingMachine

All Implemented Interfaces:

[Displayable](#), [Refillable](#), [Serializable](#), [Cloneable](#)

Direct Known Subclasses:

[CoffeeMachine](#), [SandwichMachine](#)

public class VendingMachine extends [Object](#) implements [Displayable](#), [Cloneable](#), [Serializable](#)
Represents a general vending machine.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[VendingMachine](#)([String](#) productName, int productPrice)

Constructs a new VendingMachine.

• Method Summary

All MethodsStatic MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

void

[addProduct](#)([String](#) name, int price, int quantity)

Adds a product to the vending machine.

void

[buyProduct](#)([String](#) name)

Buys a product from the vending machine.

[VendingMachine](#)

[clone](#)()

[String](#)

[displayInventory](#)()

Displays the inventory of the vending machine.

[Integer](#)

[fetchPrice](#)([String](#) name)

Fetches the price of a product.

static final int

[getMachineCount](#)()

[Map](#)<[String](#), vendingmachines.VendingMachine.Product>

[getProducts](#)()

boolean

[isEmpty](#)()

Checks if the vending machine is empty.

boolean

[needsRefill](#)()

Checks if the vending machine needs a refill.

void

[setProducts](#)([Map](#)<[String](#), vendingmachines.VendingMachine.Product> products)

Methods inherited from class java.lang.Object

equals, finalize, getClass, hashCode, notify, notifyAll, toString,
wait, wait, wait

• Constructor Details

◆ VendingMachine

public VendingMachine(String productName, int productPrice)

Constructs a new VendingMachine.

Parameters:

 productName - the name of the initial product

 productPrice - the price of the initial product

• Method Details

◆ clone

public VendingMachine clone()

Overrides:

clone in class Object

◆ getMachineCount

public static final int getMachineCount()

◆ addProduct

public void addProduct(String name, int price, int quantity)

Adds a product to the vending machine.

Parameters:

 name - the name of the product

 price - the price of the product

 quantity - the quantity of the product

◆ buyProduct

public void buyProduct(String name) throws ProductNotFoundException,
VendingMachineException

Buys a product from the vending machine.

Parameters:

 name - the name of the product

Throws:

ProductNotFoundException - if the product is not found

VendingMachineException - if the product is out of stock

◆ needsRefill

public boolean needsRefill()

Description copied from interface: Refillable

Checks if the vending machine needs a refill.

LoadTask.VendingMachineLoaderCallback

Specified by:

needsRefill in interface Refillable

Returns:

true if the vending machine needs a refill, false otherwise

♦ **isEmpty**

public boolean isEmpty()

Description copied from interface: Refillable

Checks if the vending machine is empty.

Specified by:

isEmpty in interface Refillable

Returns:

true if the vending machine is empty, false otherwise

♦ **fetchPrice**

public Integer fetchPrice(String name)

Fetches the price of a product.

Parameters:

name - the name of the product

Returns:

the price of the product, or null if not found

♦ **getProducts**

public Map<String, vendingmachines.VendingMachine.Product> getProducts()

♦ **setProducts**

public void setProducts(Map<String, vendingmachines.VendingMachine.Product> products)

♦ **displayInventory**

public String displayInventory()

Description copied from interface: Displayable

Displays the inventory of the vending machine.

Specified by:

displayInventory in interface Displayable

Returns:

the inventory as a string

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)

- Package:
- [Description](#)
- [Related Packages](#)
- [Classes and Interfaces](#)

SEARCH

Package vendingmachines

package vendingmachines

- [Classes](#)
 - [Class](#)
 - [Description](#)
 - [VendingMachine](#)
- Represents a general vending machine.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package vendingmachines

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [vendingmachines.VendingMachine](#) (implements [java.lang.Cloneable](#), interfaces [Displayable](#), [java.io.Serializable](#))

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [specializedmachines](#)

Class CoffeeMachine

[java.lang.Object](#)

[vendingmachines.VendingMachine](#)

[specializedmachines.CoffeeMachine](#)

All Implemented Interfaces:

[Displayable](#), [Refillable](#), [Serializable](#), [Cloneable](#)

public class CoffeeMachine extends [VendingMachine](#)

A specialized vending machine that dispenses coffee.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[CoffeeMachine](#)([String](#) productName, [Integer](#) productPrice, int water, int beans, int sugar)

Constructs a new CoffeeMachine with the specified initial levels of water, beans, and sugar.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[brewCoffee\(\)](#)

Brews a cup of coffee.

[String](#)

[displayInventory\(\)](#)

Displays the inventory of the vending machine.

boolean

[isEmpty\(\)](#)

Checks if the vending machine is empty.

boolean

[needsRefill\(\)](#)

Checks if the vending machine needs a refill.

void

[refillBeans](#)(int amount)

Refills the bean level.

void

[refillSugar](#)(int amount)

Refills the sugar level.

void

[refillWater](#)(int amount)

Refills the water level.

Methods inherited from class vendingmachines.VendingMachine

[addProduct](#), [buyProduct](#), [clone](#), [fetchPrice](#), [getMachineCount](#),
[getProducts](#), [setProducts](#)

Methods inherited from class java.lang.Object

[equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#),
[wait](#), [wait](#), [wait](#)

• Constructor Details

◆ CoffeeMachine

public CoffeeMachine(String productName, Integer productPrice, int water, int beans, int sugar)

Constructs a new CoffeeMachine with the specified initial levels of water, beans, and sugar.

Parameters:

productName - the name of the coffee product
 productPrice - the price of the coffee product
 water - the initial level of water
 beans - the initial level of beans
 sugar - the initial level of sugar

• Method Details

◆ brewCoffee

public void brewCoffee() throws InsufficientResourcesException

Brews a cup of coffee.

Throws:

InsufficientResourcesException - if there are not enough ingredients to brew coffee

◆ refillWater

public void refillWater(int amount)

Refills the water level.

Parameters:

amount - the amount of water to add

◆ refillBeans

public void refillBeans(int amount)

Refills the bean level.

Parameters:

amount - the amount of beans to add

◆ refillSugar

public void refillSugar(int amount)

Refills the sugar level.

Parameters:

amount - the amount of sugar to add

◆ isEmpty

public boolean isEmpty()

Description copied from interface: Refillable

Checks if the vending machine is empty.

Specified by:

isEmpty in interface Refillable

Overrides:

LoadTask.VendingMachineLoaderCallback

isEmpty in class VendingMachine

Returns:

true if the vending machine is empty, false otherwise

◆ **needsRefill**

public boolean needsRefill()

Description copied from interface: Refillable

Checks if the vending machine needs a refill.

Specified by:

needsRefill in interface Refillable

Overrides:

needsRefill in class VendingMachine

Returns:

true if the vending machine needs a refill, false otherwise

◆ **displayInventory**

public String displayInventory()

Description copied from interface: Displayable

Displays the inventory of the vending machine.

Specified by:

displayInventory in interface Displayable

Overrides:

displayInventory in class VendingMachine

Returns:

the inventory as a string

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [specializedmachines](#)

Class SandwichMachine

[java.lang.Object](#)

[vendingmachines.VendingMachine](#)

[specializedmachines.SandwichMachine](#)

All Implemented Interfaces:

[Displayable](#), [Refillable](#), [Serializable](#), [Cloneable](#)

public class SandwichMachine extends [VendingMachine](#)
A specialized vending machine that dispenses sandwiches.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[SandwichMachine](#)([String](#) productName, [Integer](#) productPrice, int ham, int cheese, int bread)

Constructs a new [SandwichMachine](#) with the specified initial quantities of ham, cheese, and bread.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

[String](#)

[displayInventory](#)()

Displays the inventory of the vending machine.

boolean

[isEmpty](#)()

Checks if the vending machine is empty.

void

[makeSandwich](#)()

Makes a sandwich.

boolean

[needsRefill](#)()

Checks if the vending machine needs a refill.

void

[refillBread](#)(int amount)

Refills the bread quantity.

void

[refillCheese](#)(int amount)

Refills the cheese quantity.

void

[refillHam](#)(int amount)

Refills the ham quantity.

Methods inherited from class vendingmachines.VendingMachine

[addProduct](#), [buyProduct](#), [clone](#), [fetchPrice](#), [getMachineCount](#),

[getProducts](#), [setProducts](#)

Methods inherited from class java.lang.Object

[equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#),

[wait](#), [wait](#), [wait](#)

• Constructor Details

◆ SandwichMachine

public SandwichMachine(String productName, Integer productPrice, int ham, int cheese, int bread)

Constructs a new SandwichMachine with the specified initial quantities of ham, cheese, and bread.

Parameters:

productName - the name of the sandwich product
 productPrice - the price of the sandwich product
 ham - the initial quantity of ham
 cheese - the initial quantity of cheese
 bread - the initial quantity of bread

• Method Details

◆ refillHam

public void refillHam(int amount)

Refills the ham quantity.

Parameters:

amount - the amount of ham to add

◆ refillCheese

public void refillCheese(int amount)

Refills the cheese quantity.

Parameters:

amount - the amount of cheese to add

◆ refillBread

public void refillBread(int amount)

Refills the bread quantity.

Parameters:

amount - the amount of bread to add

◆ makeSandwich

public void makeSandwich() throws InsufficientResourcesException

Makes a sandwich.

Throws:

InsufficientResourcesException - if there are not enough ingredients to make a sandwich

◆ isEmpty

public boolean isEmpty()

Description copied from interface: Refillable

Checks if the vending machine is empty.

Specified by:

isEmpty in interface Refillable

LoadTask.VendingMachineLoaderCallback

Overrides:

isEmpty in class VendingMachine

Returns:

true if the vending machine is empty, false otherwise

♦ **needsRefill**

public boolean needsRefill()

Description copied from interface: Refillable

Checks if the vending machine needs a refill.

Specified by:

needsRefill in interface Refillable

Overrides:

needsRefill in class VendingMachine

Returns:

true if the vending machine needs a refill, false otherwise

♦ **displayInventory**

public String displayInventory()

Description copied from interface: Displayable

Displays the inventory of the vending machine.

Specified by:

displayInventory in interface Displayable

Overrides:

displayInventory in class VendingMachine

Returns:

the inventory as a string

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)

- Package:
 - [Description](#) |
 - [Related Packages](#) |
 - [Classes and Interfaces](#)

SEARCH

Package specializedmachines

package specializedmachines

- [Classes](#)
 - [Class](#)
 - [Description](#)
 - [CoffeeMachine](#)
A specialized vending machine that dispenses coffee.
 - [SandwichMachine](#)
A specialized vending machine that dispenses sandwiches.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package specializedmachines

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [vendingmachines.VendingMachine](#) (implements [java.lang.Cloneable](#), interfaces [Displayable](#), [java.io.Serializable](#))
 - ◇ [specializedmachines.CoffeeMachine](#)
 - ◇ [specializedmachines.SandwichMachine](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [interfaces](#)

Interface Displayable

All Superinterfaces:

[Refillable](#)

All Known Implementing Classes:

[CoffeeMachine](#), [SandwichMachine](#), [VendingMachine](#)

public interface Displayable extends [Refillable](#)

An interface that represents a displayable and refillable vending machine.

• Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type

Method

Description

String

displayInventory()

Displays the inventory of the vending machine.

Methods inherited from interface interfaces.Refillable

isEmpty, needsRefill

• Method Details

◆ displayInventory

String displayInventory()

Displays the inventory of the vending machine.

Returns:

the inventory as a string

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [interfaces](#)

Interface Refillable

All Known Subinterfaces:

[Displayable](#)

All Known Implementing Classes:

[CoffeeMachine](#), [SandwichMachine](#), [VendingMachine](#)

public interface Refillable

An interface that represents a refillable vending machine.

• Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type

Method

Description

boolean

isEmpty()

Checks if the vending machine is empty.

boolean

needsRefill()

Checks if the vending machine needs a refill.

• Method Details

◆ needsRefill

boolean needsRefill()

Checks if the vending machine needs a refill.

Returns:

true if the vending machine needs a refill, false otherwise

◆ isEmpty

boolean isEmpty()

Checks if the vending machine is empty.

Returns:

true if the vending machine is empty, false otherwise

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)

- Package:
 - [Description](#) |
 - [Related Packages](#) |
 - [Classes and Interfaces](#)

SEARCH

Package interfaces

package interfaces

- Interfaces
 - Class
 - Description
 - [Displayable](#)
An interface that represents a displayable and refillable vending machine.
 - [Refillable](#)
An interface that represents a refillable vending machine.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package interfaces

Package Hierarchies:

- [All Packages](#)

Interface Hierarchy

- [interfaces.Refillable](#)
 - ◆ [interfaces.Displayable](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

A

[addProduct\(String, int, int\)](#) - Method in class vendingmachines.[VendingMachine](#)
Adds a product to the vending machine.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

B

[brewCoffee\(\)](#) - Method in class specializedmachines.[CoffeeMachine](#)

Brews a cup of coffee.

[buyProduct\(String\)](#) - Method in class vendingmachines.[VendingMachine](#)

Buys a product from the vending machine.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

C

[clone\(\)](#) - Method in class vendingmachines.[VendingMachine](#)

[CoffeeMachine](#) - Class in [specializedmachines](#)

A specialized vending machine that dispenses coffee.

[CoffeeMachine\(String, Integer, int, int, int\)](#) - Constructor for class specializedmachines.[CoffeeMachine](#)

Constructs a new CoffeeMachine with the specified initial levels of water, beans, and sugar.

[createVendingMachine\(String\)](#) - Static method in class Templates.[VendingMachineFactory](#)

Creates a vending machine of the specified type.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#)[All Packages](#)[Serialized Form](#)

D

[Displayable](#) - Interface in [interfaces](#)

An interface that represents a displayable and refillable vending machine.

[displayInventory\(\)](#) - Method in interface [interfaces.Displayable](#)

Displays the inventory of the vending machine.

[displayInventory\(\)](#) - Method in class [specializedmachines.CoffeeMachine](#)

[displayInventory\(\)](#) - Method in class [specializedmachines.SandwichMachine](#)

[displayInventory\(\)](#) - Method in class [vendingmachines.VendingMachine](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#)[All Packages](#)[Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

E

[exceptions](#) - package exceptions

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

F

[fetchPrice\(String\)](#) - Method in class vendingmachines.[VendingMachine](#)
Fetches the price of a product.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

G

[getMachineCount\(\)](#) - Static method in class vendingmachines.[VendingMachine](#)

[getOperation\(\)](#) - Method in exception class exceptions.[InsufficientResourcesException](#)
Gets the operation that caused the exception.

[getProducts\(\)](#) - Method in class vendingmachines.[VendingMachine](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#)[All Packages](#)[Serialized Form](#)

I

[InsufficientResourcesException](#) - Exception Class in [exceptions](#)

This exception is thrown when there are insufficient resources to complete an operation.

[InsufficientResourcesException\(String, String\)](#) - Constructor for exception class [exceptions](#).[InsufficientResourcesException](#)

Constructs a new exception with the specified detail message and operation.

[interfaces](#) - package interfaces

[isEmpty\(\)](#) - Method in interface [interfaces](#).[Refillable](#)

Checks if the vending machine is empty.

[isEmpty\(\)](#) - Method in class [specializedmachines](#).[CoffeeMachine](#)

[isEmpty\(\)](#) - Method in class [specializedmachines](#).[SandwichMachine](#)

[isEmpty\(\)](#) - Method in class [vendingmachines](#).[VendingMachine](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#)[All Packages](#)[Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

L

[LoadTask](#) - Class in [Unnamed Package](#)

A task that loads the state of a vending machine from a file.

[LoadTask\(String, LoadTask.VendingMachineLoaderCallback\)](#) - Constructor for class [LoadTask](#)

Constructs a new LoadTask.

[LoadTask.VendingMachineLoaderCallback](#) - Interface in [Unnamed Package](#)

Callback interface for load completion.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

M

[main\(String\[\]\)](#) - Static method in class [VendingMachineConsole](#)

[makeSandwich\(\)](#) - Method in class [specializedmachines.SandwichMachine](#)
Makes a sandwich.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

N

[needsRefill\(\)](#) - Method in interface interfaces.[Refillable](#)

Checks if the vending machine needs a refill.

[needsRefill\(\)](#) - Method in class specializedmachines.[CoffeeMachine](#)

[needsRefill\(\)](#) - Method in class specializedmachines.[SandwichMachine](#)

[needsRefill\(\)](#) - Method in class vendingmachines.[VendingMachine](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

O

[onLoadComplete\(VendingMachine\)](#) - Method in interface [LoadTask.VendingMachineLoaderCallback](#)

Called when the load is complete.

[onLoadComplete\(VendingMachine\)](#) - Method in class [VendingMachineConsole](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

P

[ProductNameAlreadyExistsException](#) - Exception Class in [exceptions](#)

This exception is thrown when a product name already exists in the vending machine.

[ProductNameAlreadyExistsException\(String\)](#) - Constructor for exception class

[exceptions.ProductNameAlreadyExistsException](#)

Constructs a new exception with the specified detail message.

[ProductNotFoundException](#) - Exception Class in [exceptions](#)

This exception is thrown when a product is not found in the vending machine.

[ProductNotFoundException\(String\)](#) - Constructor for exception class [exceptions.ProductNotFoundException](#)

Constructs a new exception with the specified detail message.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

R

[Refillable](#) - Interface in [interfaces](#)

An interface that represents a refillable vending machine.

[refillBeans\(int\)](#) - Method in class specializedmachines.[CoffeeMachine](#)

Refills the bean level.

[refillBread\(int\)](#) - Method in class specializedmachines.[SandwichMachine](#)

Refills the bread quantity.

[refillCheese\(int\)](#) - Method in class specializedmachines.[SandwichMachine](#)

Refills the cheese quantity.

[refillHam\(int\)](#) - Method in class specializedmachines.[SandwichMachine](#)

Refills the ham quantity.

[refillSugar\(int\)](#) - Method in class specializedmachines.[CoffeeMachine](#)

Refills the sugar level.

[refillWater\(int\)](#) - Method in class specializedmachines.[CoffeeMachine](#)

Refills the water level.

[run\(\)](#) - Method in class [LoadTask](#)

[run\(\)](#) - Method in class [SaveTask](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

S

[SandwichMachine](#) - Class in [specializedmachines](#)

A specialized vending machine that dispenses sandwiches.

[SandwichMachine\(String, Integer, int, int, int\)](#) - Constructor for class [specializedmachines.SandwichMachine](#)

Constructs a new [SandwichMachine](#) with the specified initial quantities of ham, cheese, and bread.

[SaveTask](#) - Class in [Unnamed Package](#)

A task that saves the state of a vending machine to a file.

[SaveTask\(String, VendingMachine\)](#) - Constructor for class [SaveTask](#)

Constructs a new [SaveTask](#).

[setProducts\(Map<String, VendingMachine.Product>\)](#) - Method in class [vendingmachines.VendingMachine](#)

[specializedmachines](#) - package [specializedmachines](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

T

[Templates](#) - package [Templates](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Index

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

V

[VendingMachine](#) - Class in [vendingmachines](#)

Represents a general vending machine.

[VendingMachine\(String, int\)](#) - Constructor for class [vendingmachines.VendingMachine](#)

Constructs a new VendingMachine.

[VendingMachineConsole](#) - Class in [Unnamed Package](#)

Console application for managing vending machines.

[VendingMachineConsole\(\)](#) - Constructor for class [VendingMachineConsole](#)

[VendingMachineException](#) - Exception Class in [exceptions](#)

Base class for exceptions thrown by the vending machine.

[VendingMachineException\(String\)](#) - Constructor for exception class [exceptions.VendingMachineException](#)

Constructs a new exception with the specified detail message.

[VendingMachineFactory](#) - Class in [Templates](#)

Factory class for creating vending machines.

[VendingMachineFactory\(\)](#) - Constructor for class [Templates.VendingMachineFactory](#)

[vendingmachines](#) - package [vendingmachines](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [V](#)

[All Classes and Interfaces](#) [All Packages](#) [Serialized Form](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [exceptions](#)

Class InsufficientResourcesException

[java.lang.Object](#)

[java.lang.Throwable](#)

[java.lang.Exception](#)

[exceptions.VendingMachineException](#)

[exceptions.InsufficientResourcesException](#)

All Implemented Interfaces:

[Serializable](#)

Class InsufficientResourcesException

public class InsufficientResourcesException extends [VendingMachineException](#)
 This exception is thrown when there are insufficient resources to complete an operation.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[InsufficientResourcesException](#)([String](#) message, [String](#) operation)
 Constructs a new exception with the specified detail message and operation.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

[String](#)

[getOperation](#)()

Gets the operation that caused the exception.

Methods inherited from class [java.lang.Throwable](#)

[addSuppressed](#), [fillInStackTrace](#), [getCause](#), [getLocalizedMessage](#),
[getMessage](#), [getStackTrace](#), [getSuppressed](#), [initCause](#),
[printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [setStackTrace](#),
[toString](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[wait](#), [wait](#), [wait](#)

• Constructor Details

◆ InsufficientResourcesException

public InsufficientResourcesException([String](#) message, [String](#) operation)
 Constructs a new exception with the specified detail message and operation.

Parameters:

message - the detail message

operation - the operation that caused the exception

• Method Details

◆ **getOperation**

public String getOperation()

Gets the operation that caused the exception.

Returns:

the operation

JavaScript is disabled on your browser.

Skip navigation links

- Overview
- Package
- Class
- Tree
- Index
- Help

- Summary:
 - ◆ Nested
 - ◆ Field
 - ◆ Constr
 - ◆ Method
- Detail:
 - ◆ Field
 - ◆ Constr
 - ◆ Method

- Summary:
 - Nested |
 - Field |
 - Constr |
 - Method

- Detail:
 - Field |
 - Constr |
 - Method

SEARCH

Package exceptions

Class ProductNameAlreadyExistsException

java.lang.Object

java.lang.Throwable

java.lang.Exception

exceptions.VendingMachineException

exceptions.ProductNameAlreadyExistsException

All Implemented Interfaces:

Serializable

Class ProductNameAlreadyExistsException

public class `ProductNameAlreadyExistsException` extends `VendingMachineException`
This exception is thrown when a product name already exists in the vending machine.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[`ProductNameAlreadyExistsException\(String message\)`](#)

Constructs a new exception with the specified detail message.

• Method Summary

Methods inherited from class `java.lang.Throwable`

[`addSuppressed`](#), [`fillInStackTrace`](#), [`getCause`](#), [`getLocalizedMessage`](#),
[`getMessage`](#), [`getStackTrace`](#), [`getSuppressed`](#), [`initCause`](#),
[`printStackTrace`](#), [`printStackTrace`](#), [`printStackTrace`](#), [`setStackTrace`](#),
[`toString`](#)

Methods inherited from class `java.lang.Object`

[`clone`](#), [`equals`](#), [`finalize`](#), [`getClass`](#), [`hashCode`](#), [`notify`](#), [`notifyAll`](#),
[`wait`](#), [`wait`](#), [`wait`](#)

• Constructor Details

◆ `ProductNameAlreadyExistsException`

public `ProductNameAlreadyExistsException(String message)`

Constructs a new exception with the specified detail message.

Parameters:

`message` - the detail message

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- Detail:
- [Field |](#)
- [Constr |](#)
- [Method](#)

SEARCH

Package [exceptions](#)

Class ProductNotFoundException

[java.lang.Object](#)

[java.lang.Throwable](#)

[java.lang.Exception](#)

[exceptions.VendingMachineException](#)

[exceptions.ProductNotFoundException](#)

All Implemented Interfaces:

[Serializable](#)

Class ProductNotFoundException

public class ProductNotFoundException extends [VendingMachineException](#)
This exception is thrown when a product is not found in the vending machine.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[ProductNotFoundException](#)([String](#) message)

Constructs a new exception with the specified detail message.

• Method Summary

Methods inherited from class [java.lang.Throwable](#)

[addSuppressed](#), [fillInStackTrace](#), [getCause](#), [getLocalizedMessage](#),
[getMessage](#), [getStackTrace](#), [getSuppressed](#), [initCause](#),
[printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [setStackTrace](#),
[toString](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[wait](#), [wait](#), [wait](#)

• Constructor Details

◆ ProductNotFoundException

public ProductNotFoundException([String](#) message)

Constructs a new exception with the specified detail message.

Parameters:

message - the detail message

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- Detail:
- [Field |](#)
- [Constr |](#)
- [Method](#)

SEARCH

Package [exceptions](#)

Class VendingMachineException

[java.lang.Object](#)

[java.lang.Throwable](#)

[java.lang.Exception](#)

[exceptions.VendingMachineException](#)

All Implemented Interfaces:

[Serializable](#)

Direct Known Subclasses:

Class [VendingMachineException](#)

[InsufficientResourcesException](#), [ProductNameAlreadyExistsException](#),
[ProductNotFoundException](#)

public class VendingMachineException extends [Exception](#)
 Base class for exceptions thrown by the vending machine.

See Also:

◊ [Serialized Form](#)

• Constructor Summary

Constructors

Constructor

Description

[VendingMachineException\(String message\)](#)

Constructs a new exception with the specified detail message.

• Method Summary

Methods inherited from class [java.lang.Throwable](#)

[addSuppressed](#), [fillInStackTrace](#), [getCause](#), [getLocalizedMessage](#),
[getMessage](#), [getStackTrace](#), [getSuppressed](#), [initCause](#),
[printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [setStackTrace](#),
[toString](#)

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[wait](#), [wait](#), [wait](#)

• Constructor Details

◆ VendingMachineException

public VendingMachineException([String](#) message)

Constructs a new exception with the specified detail message.

Parameters:

message - the detail message

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)
- Package:
- [Description |](#)
- [Related Packages |](#)
- [Classes and Interfaces](#)

SEARCH

Package exceptions

package exceptions

- Exception Classes
 - Class
 - Description
 - [InsufficientResourcesException](#)
This exception is thrown when there are insufficient resources to complete an operation.
 - [ProductNameAlreadyExistsException](#)
This exception is thrown when a product name already exists in the vending machine.
 - [ProductNotFoundException](#)
This exception is thrown when a product is not found in the vending machine.
 - [VendingMachineException](#)
Base class for exceptions thrown by the vending machine.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package exceptions

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [java.lang.Throwable](#) (implements [java.io.Serializable](#))
 - ◇ [java.lang.Exception](#)
 - [exceptions.VendingMachineException](#)
 - [exceptions.InsufficientResourcesException](#)
 - [exceptions.ProductNameAlreadyExistsException](#)
 - [exceptions.ProductNotFoundException](#)

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Package [Templates](#)

Class VendingMachineFactory

[java.lang.Object](#)

Templates.VendingMachineFactory

public abstract class VendingMachineFactory extends [Object](#)
Factory class for creating vending machines.

• Constructor Summary

Constructors
Constructor

Description

VendingMachineFactory()

• Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type

Method

Description

static VendingMachine

createVendingMachine(String type)

Creates a vending machine of the specified type.

Methods inherited from class java.lang.Object

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

• Constructor Details

♦ VendingMachineFactory

public VendingMachineFactory()

• Method Details

♦ createVendingMachine

public static VendingMachine createVendingMachine(String type) throws

IllegalArgumentException

Creates a vending machine of the specified type.

Parameters:

type - the type of vending machine

Returns:

the created vending machine

Throws:

IllegalArgumentException - if the type is unknown

JavaScript is disabled on your browser.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)

- Package:
- [Description](#)
- [Related Packages](#)
- [Classes and Interfaces](#)

SEARCH

Package Templates

package Templates

- [Classes](#)
 - [Class](#)
 - [Description](#)
 - [VendingMachineFactory](#)
 - Factory class for creating vending machines.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package Templates

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [Templates.VendingMachineFactory](#)