

Vending Machine

Generated by Doxygen 1.10.0

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 CoffeeMachine Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	7
4.1.2.1 displayType()	7
4.2 CustomOutputStream Class Reference	8
4.2.1 Detailed Description	8
4.3 CustomStreamBuffer Class Reference	8
4.3.1 Detailed Description	9
4.4 DisplayError Class Reference	9
4.4.1 Detailed Description	9
4.5 InsufficientBalance Class Reference	9
4.5.1 Detailed Description	10
4.6 InvalidValue Class Reference	10
4.6.1 Detailed Description	10
4.7 IVendingType Class Reference	10
4.7.1 Detailed Description	11
4.8 MissingProduct Class Reference	11
4.8.1 Detailed Description	11
4.9 Product Class Reference	12
4.9.1 Constructor & Destructor Documentation	12
4.9.1.1 Product()	12
4.9.2 Member Function Documentation	13
4.9.2.1 getID()	13
4.9.2.2 getInfo()	13
4.9.2.3 getName()	13
4.9.2.4 getNextID()	13
4.9.2.5 getPrice()	13
4.9.2.6 getQuantity()	14
4.9.2.7 setPrice()	14
4.9.2.8 setQuantity()	14
4.9.3 Friends And Related Symbol Documentation	14
4.9.3.1 operator<<	14
4.9.3.2 operator>>	15

4.10 SnackMachine Class Reference	15
4.10.1 Detailed Description	15
4.10.2 Member Function Documentation	16
4.10.2.1 displayType()	16
4.11 VendingMachine Class Reference	16
4.11.1 Constructor & Destructor Documentation	17
4.11.1.1 VendingMachine()	17
4.11.2 Member Function Documentation	17
4.11.2.1 addProduct()	17
4.11.2.2 displayProducts()	17
4.11.2.3 getBalance()	18
4.11.2.4 getMachinesCount()	18
4.11.2.5 getName()	18
4.11.2.6 getType()	18
4.11.2.7 insertMoney()	18
4.11.2.8 purchaseProduct()	19
4.11.2.9 setProdQuant()	19
4.11.2.10 setType()	19
4.11.3 Friends And Related Symbol Documentation	19
4.11.3.1 operator<<	19
4.11.3.2 operator>>	20
5 File Documentation	21
5.1 customParam.h File Reference	21
5.1.1 Detailed Description	21
5.2 customParam.h	22
5.3 Interface.h File Reference	22
5.3.1 Detailed Description	23
5.4 Interface.h	23
5.5 Product.cpp File Reference	23
5.5.1 Detailed Description	24
5.5.2 Function Documentation	24
5.5.2.1 operator<<()	24
5.5.2.2 operator>>()	24
5.6 Product.h File Reference	24
5.6.1 Detailed Description	25
5.7 Product.h	25
5.8 VendingMachine.cpp File Reference	25
5.8.1 Detailed Description	26
5.8.2 Function Documentation	26
5.8.2.1 operator<<()	26
5.8.2.2 operator>>()	26

5.9 VendingMachine.h File Reference	26
5.9.1 Detailed Description	27
5.10 VendingMachine.h	27
Index	29

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

exception	
DisplayError	9
InsufficientBalance	9
InvalidValue	10
MissingProduct	11
IVendingType	10
CoffeeMachine	7
SnackMachine	15
ostream	
CustomOutputStream	8
Product	12
streambuf	
CustomStreamBuffer	8
VendingMachine	16

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CoffeeMachine	Concrete interface representing Coffee machine type of a vending machine	7
CustomOutputStream	Custom output to both txt and std output	8
CustomStreamBuffer	Preparation for outputting test results to both txt and std output	8
DisplayError	No products to display exception	9
InsufficientBalance	Insufficient balance to purchase selected product	9
InvalidValue	Invalid value exception	10
IVendingType	Abstract interface for the Bridge template implementation	10
MissingProduct	Missing product exception(Invalid ID provided)	11
Product	12
SnackMachine	Concrete interface representing Snack machine type of a vending machine	15
VendingMachine	16

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

customParam.h		
	File with exceptions and custom output	21
Interface.h		
	Interfaces to implement bridge strategy	22
Product.cpp		
	Product class methods implementation	23
Product.h		
	Product class header	24
VendingMachine.cpp		
	VendingMachine class methods implementation	25
VendingMachine.h		
	VendingMachine class header	26

Chapter 4

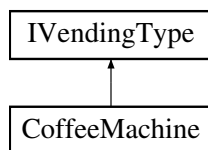
Class Documentation

4.1 CoffeeMachine Class Reference

Concrete interface representing Coffee machine type of a vending machine.

```
#include <Interface.h>
```

Inheritance diagram for CoffeeMachine:



Public Member Functions

- string [displayType](#) () override

4.1.1 Detailed Description

Concrete interface representing Coffee machine type of a vending machine.

4.1.2 Member Function Documentation

4.1.2.1 displayType()

```
string CoffeeMachine::displayType ( ) [inline], [override], [virtual]
```

Implements [IVendingType](#).

The documentation for this class was generated from the following file:

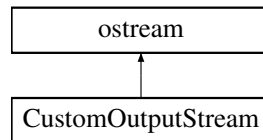
- [Interface.h](#)

4.2 CustomOutputStream Class Reference

Custom output to both txt and std output.

```
#include <customParam.h>
```

Inheritance diagram for CustomOutputStream:



Public Member Functions

- **CustomOutputStream** (const string &filename)

4.2.1 Detailed Description

Custom output to both txt and std output.

The documentation for this class was generated from the following file:

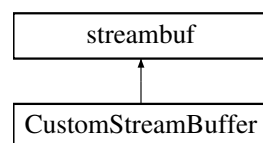
- [customParam.h](#)

4.3 CustomStreamBuffer Class Reference

Preparation for outputting test results to both txt and std output.

```
#include <customParam.h>
```

Inheritance diagram for CustomStreamBuffer:



Public Member Functions

- **CustomStreamBuffer** (const string &filename)
- virtual streamsize **xspn** (const char *s, streamsize n) override
- virtual int **overflow** (int c) override

4.3.1 Detailed Description

Preparation for outputting test results to both txt and std output.

The documentation for this class was generated from the following file:

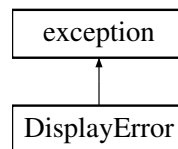
- [customParam.h](#)

4.4 DisplayError Class Reference

No products to display exception.

```
#include <customParam.h>
```

Inheritance diagram for DisplayError:



Public Member Functions

- `const char * what () const` noexcept override

4.4.1 Detailed Description

No products to display exception.

The documentation for this class was generated from the following file:

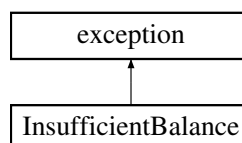
- [customParam.h](#)

4.5 InsufficientBalance Class Reference

Insufficient balance to purchase selected product.

```
#include <customParam.h>
```

Inheritance diagram for InsufficientBalance:



Public Member Functions

- `const char * what () const` noexcept override

4.5.1 Detailed Description

Insufficient balance to purchase selected product.

The documentation for this class was generated from the following file:

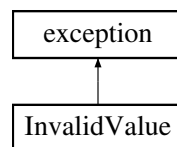
- [customParam.h](#)

4.6 InvalidValue Class Reference

Invalid value exception.

```
#include <customParam.h>
```

Inheritance diagram for InvalidValue:



Public Member Functions

- `const char * what () const` noexcept override

4.6.1 Detailed Description

Invalid value exception.

The documentation for this class was generated from the following file:

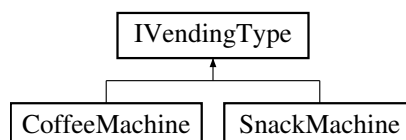
- [customParam.h](#)

4.7 IVendingType Class Reference

Abstract interface for the Bridge template implementation.

```
#include <Interface.h>
```

Inheritance diagram for IVendingType:



Public Member Functions

- virtual string **displayType** ()=0

4.7.1 Detailed Description

Abstract interface for the Bridge template implementation.

The documentation for this class was generated from the following file:

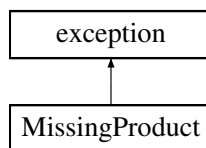
- [Interface.h](#)

4.8 MissingProduct Class Reference

Missing product exception(Invalid ID provided)

```
#include <customParam.h>
```

Inheritance diagram for MissingProduct:



Public Member Functions

- const char * **what** () const noexcept override

4.8.1 Detailed Description

Missing product exception(Invalid ID provided)

The documentation for this class was generated from the following file:

- [customParam.h](#)

4.9 Product Class Reference

Public Member Functions

- [Product](#) (string name, double price, int quantity)
Construct a new [Product](#) object.
- [~Product](#) ()
Destroy the [Product](#) object.
- string [getName](#) ()
Get the Name field.
- double [getPrice](#) ()
Get the Price field.
- int [getQuantity](#) ()
Get the Quantity field.
- int [getID](#) ()
Get products ID.
- string [getInfo](#) ()
Returns string with full info about the product.
- void [setPrice](#) (double newPrice)
Set the Price field.
- void [setQuantity](#) (int newQuantity)
Set the Quantity field.

Static Public Member Functions

- static int [getNextID](#) ()
Get the Next ID field.

Friends

- ostream & [operator<<](#) (ostream &os, const [Product](#) &product)
[Product](#) class objects serialization.
- istream & [operator>>](#) (istream &is, [Product](#) &product)
[Product](#) class objects deserialization.

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Product()

```
Product::Product (
    string name,
    double price,
    int quantity )
```

Construct a new [Product](#) object.

Parameters

<i>name</i>	
<i>price</i>	
<i>quantity</i>	

4.9.2 Member Function Documentation

4.9.2.1 getID()

```
int Product::getID ( )
```

Get products ID.

Returns

int

4.9.2.2 getInfo()

```
string Product::getInfo ( )
```

Returns string with full info about the product.

Returns

string

4.9.2.3 getName()

```
string Product::getName ( )
```

Get the Name field.

Returns

string

4.9.2.4 getNextID()

```
int Product::getNextID ( ) [static]
```

Get the Next ID field.

Returns

int

4.9.2.5 getPrice()

```
double Product::getPrice ( )
```

Get the Price field.

Returns

double

4.9.2.6 getQuantity()

```
int Product::getQuantity ( )
```

Get the Quantity field.

Returns

int

4.9.2.7 setPrice()

```
void Product::setPrice (
    double newPrice )
```

Set the Price field.

Parameters

<i>newPrice</i>	
-----------------	--

4.9.2.8 setQuantity()

```
void Product::setQuantity (
    int newQuantity )
```

Set the Quantity field.

Parameters

<i>newQuantity</i>	
--------------------	--

4.9.3 Friends And Related Symbol Documentation

4.9.3.1 operator<<

```
ostream & operator<< (
    ostream & os,
    const Product & product ) [friend]
```

[Product](#) class objects serialization.

Parameters

<i>os</i>	
<i>product</i>	

Returns

ostream&

4.9.3.2 operator>>

```
istream & operator>> (
    istream & is,
    Product & product ) [friend]
```

[Product](#) class objects deserialization.

Parameters

<i>is</i>	
<i>product</i>	

Returns

istream&

The documentation for this class was generated from the following files:

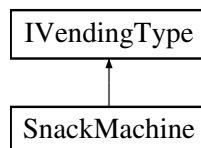
- [Product.h](#)
- [Product.cpp](#)

4.10 SnackMachine Class Reference

Concrete interface representing Snack machine type of a vending machine.

```
#include <Interface.h>
```

Inheritance diagram for SnackMachine:

**Public Member Functions**

- string [displayType](#) () override

4.10.1 Detailed Description

Concrete interface representing Snack machine type of a vending machine.

4.10.2 Member Function Documentation

4.10.2.1 displayType()

```
string SnackMachine::displayType ( ) [inline], [override], [virtual]
```

Implements [IVendingType](#).

The documentation for this class was generated from the following file:

- [Interface.h](#)

4.11 VendingMachine Class Reference

Public Member Functions

- [VendingMachine](#) (string name)
Construct a new Vending Machine object.
- [~VendingMachine](#) ()
Destroy the Vending Machine object.
- string [getType](#) ()
Get vending machines type.
- void [setType](#) ([IVendingType](#) *type)
Set vending machines type.
- string [getName](#) ()
Get the Name field.
- double [getBalance](#) ()
Get the Balance field.
- void [insertMoney](#) (double amount)
Insert money into the vending machine, increase balance.
- void [addProduct](#) (string name, double price, int quantity)
Add new product to the vending machine.
- string [displayProducts](#) ()
Display all existing products.
- void [purchaseProduct](#) (int ID)
Function to complete the purchase of the product.
- void [setProdQuant](#) (int ID, int newQuant)
Set the Prod Quant field.

Static Public Member Functions

- static int [getMachinesCount](#) ()
Get the Machines Count field.

Protected Attributes

- [IVendingType](#) * [vendingType](#)

Friends

- ostream & [operator<<](#) (ostream &os, const [VendingMachine](#) &vendingMachine)
Class serialization.
- istream & [operator>>](#) (istream &is, [VendingMachine](#) &vendingMachine)
Class deserialization.

4.11.1 Constructor & Destructor Documentation

4.11.1.1 VendingMachine()

```
VendingMachine::VendingMachine (
    string name )
```

Construct a new Vending Machine object.

Parameters

<i>name</i>	
-------------	--

4.11.2 Member Function Documentation

4.11.2.1 addProduct()

```
void VendingMachine::addProduct (
    string name,
    double price,
    int quantity )
```

Add new product to the vending machine.

Parameters

<i>name</i>	
<i>price</i>	
<i>quantity</i>	

4.11.2.2 displayProducts()

```
string VendingMachine::displayProducts ( )
```

Display all existing products.

Returns

string

4.11.2.3 getBalance()

```
double VendingMachine::getBalance ( )
```

Get the Balance field.

Returns

double

4.11.2.4 getMachinesCount()

```
int VendingMachine::getMachinesCount ( ) [static]
```

Get the Machines Count field.

Returns

int

4.11.2.5 getName()

```
string VendingMachine::getName ( )
```

Get the Name field.

Returns

string

4.11.2.6 getType()

```
string VendingMachine::getType ( )
```

Get vending machines type.

Returns

string

4.11.2.7 insertMoney()

```
void VendingMachine::insertMoney (
    double amount )
```

Insert money into the vending machine, increase balance.

Parameters

<i>amount</i>	
---------------	--

4.11.2.8 purchaseProduct()

```
void VendingMachine::purchaseProduct (
    int ID )
```

Function to complete the purchase of the product.

Parameters

<i>ID</i>	
-----------	--

4.11.2.9 setProdQuant()

```
void VendingMachine::setProdQuant (
    int ID,
    int newQuant )
```

Set the Prod Quant field.

Parameters

<i>ID</i>	
<i>newQuant</i>	

4.11.2.10 setType()

```
void VendingMachine::setType (
    IVendingType * type )
```

Set vending machines type.

Parameters

<i>type</i>	
-------------	--

4.11.3 Friends And Related Symbol Documentation**4.11.3.1 operator<<**

```
ostream & operator<< (
    ostream & os,
    const VendingMachine & vendingMachine ) [friend]
```

Class serialization.

Parameters

<i>os</i>	
<i>vendingMachine</i>	

Returns

ostream&

4.11.3.2 operator>>

```
istream & operator>> (  
    istream & is,  
    VendingMachine & vendingMachine ) [friend]
```

Class deserialization.

Parameters

<i>is</i>	
<i>vendingMachine</i>	

Returns

istream&

The documentation for this class was generated from the following files:

- [VendingMachine.h](#)
- [VendingMachine.cpp](#)

Chapter 5

File Documentation

5.1 customParam.h File Reference

File with exceptions and custom output.

```
#include <iostream>
#include <exception>
#include <string>
```

Classes

- class [MissingProduct](#)
Missing product exception(Invalid ID provided)
- class [DisplayError](#)
No products to display exception.
- class [InvalidValue](#)
Invalid value exception.
- class [InsufficientBalance](#)
Insufficient balance to purchase selected product.
- class [CustomStreamBuffer](#)
Preparation for outputting test results to both txt and std output.
- class [CustomOutputStream](#)
Custom output to both txt and std output.

5.1.1 Detailed Description

File with exceptions and custom output.

Author

Leonardas Sinkevicius

5.2 customParam.h

[Go to the documentation of this file.](#)

```

00001
00007 #ifndef CUSTOMPARAM_H
00008 #define CUSTOMPARAM_H
00009
00010 #include <iostream>
00011 #include <exception>
00012 #include <string>
00013 using namespace std;
00018 class MissingProduct : public exception {
00019 public:
00020     const char* what() const noexcept override {
00021         return "Exception: Missing product.";
00022     }
00023 };
00024
00029 class DisplayError : public exception {
00030 public:
00031     const char* what() const noexcept override {
00032         return "Exception: Nothing to display.";
00033     }
00034 };
00035
00040 class InvalidValue : public exception {
00041 public:
00042     const char* what() const noexcept override {
00043         return "Exception: Invalid value is provided.";
00044     }
00045 };
00046
00051 class InsufficientBalance : public exception {
00052 public:
00053     const char* what() const noexcept override {
00054         return "Exception: Insufficient balance.";
00055     }
00056 };
00057
00062 class CustomStreamBuffer : public streambuf {
00063 public:
00064     CustomStreamBuffer(const string& filename) : file(filename) {}
00065
00066     virtual streamsize xsputn(const char* s, streamsize n) override {
00067         // Write to the file
00068         file.write(s, n);
00069         file.flush(); // Flush the file stream
00070         // Write to the standard output
00071         return cout.rdbuf()->sputn(s, n);
00072     }
00073
00074     virtual int overflow(int c) override {
00075         // Write to the file
00076         file.put(c);
00077         file.flush(); // Flush the file stream
00078         // Write to the standard output
00079         return cout.rdbuf()->sputc(c);
00080     }
00081
00082 private:
00083     ofstream file;
00084 };
00085
00090 class CustomOutputStream : public ostream {
00091 public:
00092     CustomOutputStream(const string& filename) : ostream(&buffer), buffer(filename) {}
00093
00094 private:
00095     CustomStreamBuffer buffer;
00096 };
00097
00098 #endif

```

5.3 Interface.h File Reference

Interfaces to implement bridge strategy.

Classes

- class [IVendingType](#)
Abstract interface for the Bridge template implementation.
- class [CoffeeMachine](#)
Concrete interface representing Coffee machine type of a vending machine.
- class [SnackMachine](#)
Concrete interface representing Snack machine type of a vending machine.

5.3.1 Detailed Description

Interfaces to implement bridge strategy.

Author

Leonardas Sinkevicius

5.4 Interface.h

[Go to the documentation of this file.](#)

```
00001
00007 using namespace std;
00008
00013 class IVendingType{
00014     public:
00015         virtual string displayType() = 0;
00016         virtual ~IVendingType() {}
00017 };
00018
00023 class CoffeeMachine : public IVendingType{
00024     public:
00025         string displayType() override {
00026             return "This is Coffee machine.\n";
00027         }
00028 };
00029
00034 class SnackMachine : public IVendingType{
00035     public:
00036         string displayType() override {
00037             return "This is Snack machine.\n";
00038         }
00039 };
```

5.5 Product.cpp File Reference

[Product](#) class methods implementation.

```
#include "Product.h"
```

Functions

- ostream & [operator<<](#) (ostream &os, const [Product](#) &product)
- istream & [operator>>](#) (istream &is, [Product](#) &product)

5.5.1 Detailed Description

[Product](#) class methods implementation.

Author

Leonardas Sinkevicius

5.5.2 Function Documentation

5.5.2.1 `operator<<()`

```
ostream & operator<< (
    ostream & os,
    const Product & product )
```

Parameters

<i>os</i>	
<i>product</i>	

Returns

ostream&

5.5.2.2 `operator>>()`

```
istream & operator>> (
    istream & is,
    Product & product )
```

Parameters

<i>is</i>	
<i>product</i>	

Returns

istream&

5.6 Product.h File Reference

[Product](#) class header.

```
#include <memory>
#include <iostream>
#include <string>
#include <sstream>
```

Classes

- class [Product](#)

5.6.1 Detailed Description

[Product](#) class header.

Author

Leonardas Sinkevicius

5.7 Product.h

[Go to the documentation of this file.](#)

```

00001
00006 #ifndef PRODUCT_H
00007 #define PRODUCT_H
00008
00009 #include <memory>
00010 #include <iostream>
00011 #include <string>
00012 #include <sstream>
00013 using namespace std;
00014
00015 class Product{
00016     private:
00017         string name;
00018         double price;
00019         int quantity;
00020         int id;
00021         static int nextID;
00022     public:
00030         Product(string name, double price, int quantity);
00035         ~Product();
00036
00044         friend ostream& operator<<(ostream& os, const Product& product);
00052         friend istream& operator>>(istream& is, Product& product);
00053
00059         string getName();
00065         double getPrice();
00071         int getQuantity();
00077         int getID();
00083         static int getNextID();
00089         string getInfo();
00095         void setPrice(double newPrice);
00101         void setQuantity(int newQuantity);
00102 };
00103
00104 #endif

```

5.8 VendingMachine.cpp File Reference

[VendingMachine](#) class methods implementation.

```

#include "VendingMachine.h"
#include "customParam.h"
#include "Product.h"

```

Functions

- ostream & [operator<<](#) (ostream &os, const [VendingMachine](#) &vendingMachine)
- istream & [operator>>](#) (istream &is, [VendingMachine](#) &vendingMachine)

5.8.1 Detailed Description

[VendingMachine](#) class methods implementation.

Author

Leonardas Sinkevicius

5.8.2 Function Documentation

5.8.2.1 `operator<<()`

```
ostream & operator<< (
    ostream & os,
    const VendingMachine & vendingMachine )
```

Parameters

<i>os</i>	
<i>vendingMachine</i>	

Returns

ostream&

5.8.2.2 `operator>>()`

```
istream & operator>> (
    istream & is,
    VendingMachine & vendingMachine )
```

Parameters

<i>is</i>	
<i>vendingMachine</i>	

Returns

istream&

5.9 VendingMachine.h File Reference

[VendingMachine](#) class header.

```
#include <iostream>
#include <string>
```



```
#include <sstream>
#include <map>
#include <fstream>
#include "customParam.h"
#include "Interface.h"
```

Classes

- class [VendingMachine](#)

5.9.1 Detailed Description

[VendingMachine](#) class header.

Author

Leonardas Sinkevicius

5.10 VendingMachine.h

[Go to the documentation of this file.](#)

```
00001
00007 #ifndef VENDINGMACHINE_H
00008 #define VENDINGMACHINE_H
00009
00010 #include <iostream>
00011 #include <string>
00012 #include <sstream>
00013 #include <map>
00014 #include <fstream>
00015 #include "customParam.h"
00016 #include "Interface.h"
00017
00018 using namespace std;
00019
00020 class Product;
00021
00022 class VendingMachine{
00023
00024     private:
00025         string name;
00026         double balance;
00027         map<int, Product> products;
00028         static int machinesCount;
00029
00030     protected:
00031         IVendingType* vendingType;
00032
00033     public:
00034         VendingMachine(string name);
00035         ~VendingMachine();
00036
00037         friend ostream& operator<<(ostream& os, const VendingMachine& vendingMachine);
00038         friend istream& operator>>(istream& is, VendingMachine& vendingMachine);
00039
00040         string getType();
00041         void setType(IVendingType* type);
00042
00043         static int getMachinesCount();
00044         string getName();
00045         double getBalance();
00046         void insertMoney(double amount);
00047         void addProduct(string name, double price, int quantity);
00048         string displayProducts();
00049         void purchaseProduct(int ID);
00050         void setProdQuant(int ID, int newQuant);
00051 };
00052 #endif
```


Index

- addProduct
 - VendingMachine, 17
- CoffeeMachine, 7
 - displayType, 7
- CustomOutputStream, 8
- customParam.h, 21
- CustomStreamBuffer, 8
- DisplayError, 9
- displayProducts
 - VendingMachine, 17
- displayType
 - CoffeeMachine, 7
 - SnackMachine, 16
- getBalance
 - VendingMachine, 17
- getID
 - Product, 13
- getInfo
 - Product, 13
- getMachinesCount
 - VendingMachine, 18
- getName
 - Product, 13
 - VendingMachine, 18
- getNextID
 - Product, 13
- getPrice
 - Product, 13
- getQuantity
 - Product, 13
- getType
 - VendingMachine, 18
- insertMoney
 - VendingMachine, 18
- InsufficientBalance, 9
- Interface.h, 22
- InvalidValue, 10
- IVendingType, 10
- MissingProduct, 11
- operator<<
 - Product, 14
 - Product.cpp, 24
 - VendingMachine, 19
 - VendingMachine.cpp, 26
- operator>>
 - Product, 15
 - Product.cpp, 24
 - VendingMachine, 20
 - VendingMachine.cpp, 26
- Product, 12
 - getID, 13
 - getInfo, 13
 - getName, 13
 - getNextID, 13
 - getPrice, 13
 - getQuantity, 13
 - operator<<, 14
 - operator>>, 15
 - Product, 12
 - setPrice, 14
 - setQuantity, 14
- Product.cpp, 23
 - operator<<, 24
 - operator>>, 24
- Product.h, 24
- purchaseProduct
 - VendingMachine, 19
- setPrice
 - Product, 14
- setProdQuant
 - VendingMachine, 19
- setQuantity
 - Product, 14
- setType
 - VendingMachine, 19
- SnackMachine, 15
 - displayType, 16
- VendingMachine, 16
 - addProduct, 17
 - displayProducts, 17
 - getBalance, 17
 - getMachinesCount, 18
 - getName, 18
 - getType, 18
 - insertMoney, 18
 - operator<<, 19
 - operator>>, 20
 - purchaseProduct, 19
 - setProdQuant, 19
 - setType, 19
 - VendingMachine, 17
- VendingMachine.cpp, 25

operator<<, [26](#)
operator>>, [26](#)
VendingMachine.h, [26](#)