LESI_TP_POO_27967

v 1.0

Generated by Doxygen 1.9.8

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 trabalhoPOO_27967 Namespace Reference

Namespaces

- namespace Store

Classes

- class BestSale
- class Campaign

  Purpose: Definition of Campaign and methods to deal with Campaign operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:43 AM.
- class Campaigns

  Purpose:This file has the definition and methods to work with the plurality of Campaign. Created by: Jose Alves a27967 Created on: 11/14/2024 3:57:04 PM.
- class Categories

  Purpose:This file has the definition and methods to work with the plurality of Category. Created by: Jose Alves a27967 Created on: 11/14/2024 4:45:58 PM.
- class Category

  Purpose: Definition of Category and methods to deal with Category operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:20:47 AM.
- class Client

  Purpose: Definition of Client and methods to deal with Client operations. Created by: Jose Alves a27967 Created on: 10/29/2024 4:23:56 PM.
- class Clients

  Purpose: Class with the definition and methods to manage a list of clients. Created by: Jose Alves a27967 Created on: 11/12/2024 9:25:28 PM.
- class Make

  Purpose: Definition of Make and methods to deal with Make operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:22:09 AM.
- class Makes

  Purpose:This file has the definition and methods to work with the plurality of Make. Created by: Jose Alves a27967 Created on: 11/14/2024 4:33:51 PM.
- class Product

  Purpose: Definition of product and methods to deal with product operations. Created by: Jose Alves a27967 Created on: 11/2/2024 4:40:12 PM.

- class Products

  Purpose: Class to manage a group of more than one product. Created by: Jose Alves a27967 Created on: 11/9/2024 6:34:19 PM.
- class Sale

  Purpose: Definition of Sale and methods to deal with Sale operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:53 AM.
- class Sales

  Purpose: Class with the agregation of sales of a store. Created by: Jose Alves a27967 Created on: 11/10/2024 7:42:03 PM.
- class Warranties

  Purpose:This file has the definition and methods to work with the plurality of Warranty. Created by: Jose Alves a27967 Created on: 11/14/2024 4:20:11 PM.
- class Warranty

  Purpose: This class contains the definition and methods to manage warranties. Created by: Jose Alves a27967 Created on: 11/13/2024 4:17:18 PM.

4.2 trabalhoPOO_27967.Store Namespace Reference

Classes

- class Store

  Purpose: This class has the definition and properties to manage a store. Created by: Jose Alves a27967 Created on: 11/14/2024 5:01:23 PM.

Chapter 5

Class Documentation

5.1 trabalhoPOO_27967.BestSale Class Reference

5.1.1 Detailed Description

Definition at line 10 of file BestSale.cs.

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

- BestSale.cs

5.2 trabalhoPOO_27967.Campaign Class Reference

Purpose: Definition of Campaign and methods to deal with Campaign operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:43 AM.

Public Member Functions

- Campaign ()

  The default Constructor.
- Campaign (string name, decimal discount, DateTime startDate, DateTime endDate)

  Constructor used to create a campaign when a name, a decimal value of discount, start and ending dates are given.
- override bool Equals (object obj)

  Redefine the Equals operator to verify if a campaign matches the other.

Static Public Member Functions

- static bool operator== (Campaign camp1, Campaign camp2)

  Redefinition of the == operator.
- static bool operator!= (Campaign camp1, Campaign camp2)

  Redefinition of the != operator.
- static bool VerifyApplicability (Campaign camp)

  Method to verify if a certaign campaign is applicable, according to its start and end dates.

Properties

- int Id `[get, set]`

  Property used to get and set the ID of a Campaign.
- string Name `[get, set]`

  Property used to get and set the Name of a Campaign.
- decimal Discount `[get, set]`

  Property used to get and set the Decimal Discount of a Campaign.
- DateTime StartDate `[get, set]`

  Property used to get and set the Start Date of a Campaign.
- DateTime EndDate `[get, set]`

  Property used to get and set the End Date of a Campaign.
- int CampaignCount `[get, set]`

  Property used to get and set the Campaign Counter.

5.2.1 Detailed Description

Purpose: Definition of Campaign and methods to deal with Campaign operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:43 AM.

Definition at line 20 of file Campaign.cs.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Campaign() `[1/2]`

```
trabalhoPOO_27967.Campaign.Campaign ( )
```

The default Constructor.

Definition at line 38 of file Campaign.cs.

5.2.2.2 Campaign() `[2/2]`

```
trabalhoPOO_27967.Campaign.Campaign (
             string name,
             decimal discount,
             DateTime startDate,
             DateTime endDate )
```

Constructor used to create a campaign when a name, a decimal value of discount, start and ending dates are given.

Parameters

id	
name	
discount	
startDate	
endDate	

Definition at line 55 of file Campaign.cs.

5.2.3 Member Function Documentation

5.2.3.1 Equals()

```
override bool trabalhoPOO_27967.Campaign.Equals (
             object obj )
```

Redefine the Equals operator to verify if a campaign matches the other.

Parameters

obj	

Returns

Veriffies if the object given is null.

Casts the object to be Campaign.

Definition at line 134 of file Campaign.cs.

5.2.3.2 operator"!=()

```
static bool trabalhoPOO_27967.Campaign.operator!= (
             Campaign camp1,
             Campaign camp2 )  [static]
```

Redefinition of the != operator.

Parameters

cli1	
cli2	

Returns

Definition at line 164 of file Campaign.cs.

5.2.3.3 operator==()

```
static bool trabalhoPOO_27967.Campaign.operator== (
             Campaign camp1,
             Campaign camp2 )  [static]
```

Redefinition of the == operator.

Parameters

cli1	
cli2	

Returns

Definition at line 153 of file Campaign.cs.

5.2.3.4 VerifyApplicability()

```
static bool trabalhoPOO_27967.Campaign.VerifyApplicability (
            Campaign camp )  [static]
```

Method to verify if a certaign campaign is applicable, according to its start and end dates.

Parameters

camp	

Returns

Verifies if the actual date is in between the start and end dates of the campaign.

Definition at line 177 of file Campaign.cs.

5.2.4 Property Documentation

5.2.4.1 CampaignCount

```
int trabalhoPOO_27967.Campaign.CampaignCount  [get], [set]
```

Property used to get and set the Campaign Counter.

Definition at line 119 of file Campaign.cs.

5.2.4.2 Discount

```
decimal trabalhoPOO_27967.Campaign.Discount  [get], [set]
```

Property used to get and set the Decimal Discount of a Campaign.

Definition at line 92 of file Campaign.cs.

5.2.4.3 EndDate

```
DateTime trabalhoPOO_27967.Campaign.EndDate  [get], [set]
```

Property used to get and set the End Date of a Campaign.

Definition at line 110 of file Campaign.cs.

5.2.4.4 Id

```
int trabalhoPOO_27967.Campaign.Id  [get], [set]
```

Property used to get and set the ID of a Campaign.

Definition at line 75 of file Campaign.cs.

5.2.4.5 Name

```
string trabalhoPOO_27967.Campaign.Name  [get], [set]
```

Property used to get and set the Name of a Campaign.

Definition at line 83 of file Campaign.cs.

5.2.4.6 StartDate

```
DateTime trabalhoPOO_27967.Campaign.StartDate  [get], [set]
```

Property used to get and set the Start Date of a Campaign.

Definition at line 101 of file Campaign.cs.

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

- Campaign/Campaign.cs

5.3 trabalhoPOO_27967.Campaigns Class Reference

Purpose:This file has the definition and methods to work with the plurality of Campaign. Created by: Jose Alves a27967 Created on: 11/14/2024 3:57:04 PM.

Public Member Functions

- Campaigns ()

  The default Constructor.
- Campaigns (List< Campaign > p)

  The constructor to use when a list of Campaigns is given.
- bool AddCampaign (Campaign c)

  Method used to add a campaign to a list of campaigns.
- bool RemoveCampaign (Campaign camp)

  Method used to remove a campaign from a list of campaigns.
- bool ExistCampaign (int id)

  Method used to confirm if a campaign exists on a list of campaigns.

Properties

- List< Campaign > Camps `[get, set]`

  Property used to get and set a Campaign list.

5.3.1 Detailed Description

Purpose:This file has the definition and methods to work with the plurality of Campaign. Created by: Jose Alves a27967 Created on: 11/14/2024 3:57:04 PM.

Definition at line 22 of file Campaigns.cs.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Campaigns() `[1/2]`

```
trabalhoPOO_27967.Campaigns.Campaigns ( )
```

The default Constructor.

Definition at line 35 of file Campaigns.cs.

5.3.2.2 Campaigns() `[2/2]`

```
trabalhoPOO_27967.Campaigns.Campaigns (
            List< Campaign > p )
```

The constructor to use when a list of Campaigns is given.

Parameters

p	

Definition at line 44 of file Campaigns.cs.

5.3.3 Member Function Documentation

5.3.3.1 AddCampaign()

```
bool trabalhoPOO_27967.Campaigns.AddCampaign (
             Campaign c )
```

Method used to add a campaign to a list of campaigns.

Parameters

c	

Returns

Definition at line 73 of file Campaigns.cs.

5.3.3.2 ExistCampaign()

```
bool trabalhoPOO_27967.Campaigns.ExistCampaign (
             int id )
```

Method used to confirm if a campaign exists on a list of campaigns.

Parameters

id	

Returns

Definition at line 103 of file Campaigns.cs.

5.3.3.3 RemoveCampaign()

```
bool trabalhoPOO_27967.Campaigns.RemoveCampaign (
             Campaign camp )
```

Method used to remove a campaign from a list of campaigns.

Parameters

camp	

Returns

Definition at line 85 of file Campaigns.cs.

5.3.4 Property Documentation

5.3.4.1 Camps

```
List<Campaign> trabalhoPOO_27967.Campaigns.Camps  [get], [set]
```

Property used to get and set a Campaign list.

Definition at line 54 of file Campaigns.cs.

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

- Campaign/Campaigns.cs

5.4 trabalhoPOO_27967.Categories Class Reference

Purpose:This file has the definition and methods to work with the plurality of Category. Created by: Jose Alves a27967 Created on: 11/14/2024 4:45:58 PM.

Public Member Functions

- Categories ()

  The default Constructor.
- Categories (List< Category > cats)

  The constructor to use when a list of categories is given.
- bool AddCategory (Category c)

  Method used to add a category to a list of categories.
- bool RemoveCategory (Category c)

  Method used to remove a category from a list of categories.
- bool ExistCategory (int id)

  Method used to verify if a category exists on a list of makes, given its ID.
- bool ExistCategory (string name)

  Method used to verify if a Category exists on a list of categories, given its name.

Properties

- List< Category > Cats `[get, set]`

  The property used to get and set the list of categories.

5.4.1 Detailed Description

Purpose:This file has the definition and methods to work with the plurality of Category. Created by: Jose Alves a27967 Created on: 11/14/2024 4:45:58 PM.

Definition at line 21 of file Categories.cs.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Categories() [1/2]

```
trabalhoPOO_27967.Categories.Categories ( )
```

The default Constructor.

Definition at line 34 of file Categories.cs.

5.4.2.2 Categories() [2/2]

```
trabalhoPOO_27967.Categories.Categories (
            List< Category > cats )
```

The constructor to use when a list of categories is given.

Parameters

cats	

Definition at line 43 of file Categories.cs.

5.4.3 Member Function Documentation

5.4.3.1 AddCategory()

```
bool trabalhoPOO_27967.Categories.AddCategory (
            Category c )
```

Method used to add a category to a list of categories.

Parameters

c	

Returns

Definition at line 73 of file Categories.cs.

5.4.3.2 ExistCategory() [1/2]

```
bool trabalhoPOO_27967.Categories.ExistCategory (
             int id )
```

Method used to verify if a category exists on a list of makes, given its ID.

Parameters

id	

Returns

Definition at line 103 of file Categories.cs.

5.4.3.3 ExistCategory() [2/2]

```
bool trabalhoPOO_27967.Categories.ExistCategory (
             string name )
```

Method used to verify if a Category exists on a list of categories, given its name.

Parameters

name	

Returns

Definition at line 120 of file Categories.cs.

5.4.3.4 RemoveCategory()

```
bool trabalhoPOO_27967.Categories.RemoveCategory (
             Category c )
```

Method used to remove a category from a list of categories.

Parameters

c	

Returns

Definition at line 85 of file Categories.cs.

5.4.4 Property Documentation

5.4.4.1 Cats

```
List<Category> trabalhoPOO_27967.Categories.Cats  [get], [set]
```

The property used to get and set the list of categories.

Definition at line 55 of file Categories.cs.

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

- Category/Categories.cs

5.5 trabalhoPOO_27967.Category Class Reference

Purpose: Definition of Category and methods to deal with Category operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:20:47 AM.

Public Member Functions

- Category ()

  The default Constructor.
- Category (string name)

  The constructor to use when the name of a category is given.
- override bool Equals (object obj)

  Method that overrides Equals() and verifies if a categry matches another one.

Static Public Member Functions

- static bool operator== (Category cat1, Category cat2)

  Redefinition of the Equal operator.
- static bool operator!= (Category cat1, Category cat2)

  Redefinition of the Not Equal Operator.

Properties

- int Id `[get, set]`

  The property to get and set the ID of a category.
- string Name `[get, set]`

  The property to get and set the name of a Category.

5.5.1 Detailed Description

Purpose: Definition of Category and methods to deal with Category operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:20:47 AM.

Definition at line 20 of file Category.cs.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Category() [1/2]

```
trabalhoPOO_27967.Category.Category ( )
```

The default Constructor.

Definition at line 35 of file Category.cs.

5.5.2.2 Category() [2/2]

```
trabalhoPOO_27967.Category.Category (
             string name )
```

The constructor to use when the name of a category is given.

Parameters

name	

Definition at line 45 of file Category.cs.

5.5.3 Member Function Documentation

5.5.3.1 Equals()

```
override bool trabalhoPOO_27967.Category.Equals (
             object obj )
```

Method that overrides Equals() and verifies if a categry matches another one.

Parameters

obj	

Returns

Veriffies if the object given is null.

Casts the object to be Category.

Definition at line 81 of file Category.cs.

5.5.3.2 operator"!=()

```
static bool trabalhoPOO_27967.Category.operator!= (
            Category cat1,
            Category cat2 )  [static]
```

Redefinition of the Not Equal Operator.

Parameters

cat1	
cat2	

Returns

Definition at line 111 of file Category.cs.

5.5.3.3 operator==()

```
static bool trabalhoPOO_27967.Category.operator== (
            Category cat1,
            Category cat2 )  [static]
```

Redefinition of the Equal operator.

Parameters

cat1	
cat2	

Returns

Definition at line 100 of file Category.cs.

5.5.4 Property Documentation

5.5.4.1 Id

```
int trabalhoPOO_27967.Category.Id  [get], [set]
```

The property to get and set the ID of a category.

Definition at line 57 of file Category.cs.

5.5.4.2 Name

```
string trabalhoPOO_27967.Category.Name  [get], [set]
```

The property to get and set the name of a Category.

Definition at line 66 of file Category.cs.

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

- Category/Category.cs

5.6 trabalhoPOO_27967.Client Class Reference

Purpose: Definition of Client and methods to deal with Client operations. Created by: Jose Alves a27967 Created on: 10/29/2024 4:23:56 PM.

Public Member Functions

- Client ()

  The default Constructor.
- Client (string n, string c)

  Constructor to use when a name and a contact are given.
- override string ToString ()

  Redefine the ToString Function to show a client's info.
- override bool Equals (object obj)

  Redefine the Equals operator to verify if a client matches the other.

Static Public Member Functions

- static bool operator== (Client cli1, Client cli2)

  Redefinition of the == operator.
- static bool operator!= (Client cli1, Client cli2)

  Redefinition of the != operator.

Properties

- int ClientID `[get, set]`

  Property that sets or returns the ID of a client.
- string Name `[get, set]`

  Property that sets or returns the Name of a client.
- string Contact `[get, set]`

  Property that sets or returns the Contact of a client.
- static int ClientCount `[get, set]`

  Property that sets or returns the amount of clients.

5.6.1 Detailed Description

Purpose: Definition of Client and methods to deal with Client operations. Created by: Jose Alves a27967 Created on: 10/29/2024 4:23:56 PM.

Definition at line 23 of file Client.cs.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Client() [1/2]

```
trabalhoPOO_27967.Client.Client ( )
```

The default Constructor.

Definition at line 39 of file Client.cs.

5.6.2.2 Client() [2/2]

```
trabalhoPOO_27967.Client.Client (
             string n,
             string c )
```

Constructor to use when a name and a contact are given.

Parameters

n	
c	

Definition at line 51 of file Client.cs.

5.6.3 Member Function Documentation

5.6.3.1 Equals()

```
override bool trabalhoPOO_27967.Client.Equals (
             object obj )
```

Redefine the Equals operator to verify if a client matches the other.

Parameters

obj	

Returns

Veriffies if the object given is null.

Casts the object to be Client.

Definition at line 131 of file Client.cs.

5.6.3.2 operator"!=()

```
static bool trabalhoPOO_27967.Client.operator!= (
            Client cli1,
            Client cli2 )  [static]
```

Redefinition of the != operator.

Parameters

cli1	
cli2	

Returns

Definition at line 161 of file Client.cs.

5.6.3.3 operator==()

```
static bool trabalhoPOO_27967.Client.operator== (
            Client cli1,
            Client cli2 )  [static]
```

Redefinition of the == operator.

Parameters

cli1	
cli2	

Returns

Definition at line 150 of file Client.cs.

5.6.3.4 ToString()

```
override string trabalhoPOO_27967.Client.ToString ( )
```

Redefine the ToString Function to show a client's info.

Returns

Definition at line 116 of file Client.cs.

5.6.4 Property Documentation

5.6.4.1 ClientCount

```
int trabalhoPOO_27967.Client.ClientCount  [static], [get], [set]
```

Property that sets or returns the amount of clients.

Definition at line 100 of file Client.cs.

5.6.4.2 ClientID

```
int trabalhoPOO_27967.Client.ClientID  [get], [set]
```

Property that sets or returns the ID of a client.

Definition at line 65 of file Client.cs.

5.6.4.3 Contact

```
string trabalhoPOO_27967.Client.Contact  [get], [set]
```

Property that sets or returns the Contact of a client.

Definition at line 83 of file Client.cs.

5.6.4.4 Name

```
string trabalhoPOO_27967.Client.Name  [get], [set]
```

Property that sets or returns the Name of a client.

Definition at line 74 of file Client.cs.

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

- Client/Client.cs

5.7 trabalhoPOO_27967.Clients Class Reference

Purpose: Class with the definition and methods to manage a list of clients. Created by: Jose Alves a27967 Created on: 11/12/2024 9:25:28 PM.

Public Member Functions

- Clients ()

  The default Constructor.
- bool AddClient (Client cli)

  Method to add a client to the store's client list.
- bool RemoveClient (Client cli)

  Method to remove a client from the store's client list.
- bool ExistClient (int id)

  Method to check if a client is listed on a clients' list.
- Client GetClient (int id)

  Method used to get a client from a clients' list.

Properties

- List< Client > ClientList `[get, set]`

  Property used to get and set the client list.

5.7.1 Detailed Description

Purpose: Class with the definition and methods to manage a list of clients. Created by: Jose Alves a27967 Created on: 11/12/2024 9:25:28 PM.

Definition at line 21 of file Clients.cs.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 Clients()

```
trabalhoPOO_27967.Clients.Clients ( )
```

The default Constructor.

Definition at line 34 of file Clients.cs.

5.7.3 Member Function Documentation

5.7.3.1 AddClient()

```
bool trabalhoPOO_27967.Clients.AddClient (
            Client cli )
```

Method to add a client to the store's client list.

Parameters

cli	

Returns

Definition at line 65 of file Clients.cs.

5.7.3.2 ExistClient()

```
bool trabalhoPOO_27967.Clients.ExistClient (
             int id )
```

Method to check if a client is listed on a clients' list.

Parameters

id	

Returns

Definition at line 95 of file Clients.cs.

5.7.3.3 GetClient()

```
Client trabalhoPOO_27967.Clients.GetClient (
             int id )
```

Method used to get a client from a clients' list.

Parameters

id	

Returns

Definition at line 112 of file Clients.cs.

5.7.3.4 RemoveClient()

```
bool trabalhoPOO_27967.Clients.RemoveClient (
             Client cli )
```

Method to remove a client from the store's client list.

Parameters

cli	

Returns

Definition at line 77 of file Clients.cs.

5.7.4 Property Documentation

5.7.4.1 ClientList

```
List<Client> trabalhoPOO_27967.Clients.ClientList  [get], [set]
```

Property used to get and set the client list.

Definition at line 47 of file Clients.cs.

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

- Client/Clients.cs

5.8 trabalhoPOO_27967.Make Class Reference

Purpose: Definition of Make and methods to deal with Make operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:22:09 AM.

Public Member Functions

- Make ()

  The default Constructor.
- Make (string name)

  Constructor when the name of the make is given.
- override string ToString ()

  Override of the ToString() method to convert the data of a Make to a string.
- override bool Equals (object obj)

  The redefinition of the Equals() Method, to verify if a make matches another one.

Static Public Member Functions

- static bool operator== (Make m1, Make m2)

  The redefinition of the Equal operator.
- static bool operator!= (Make m1, Make m2)

  The redefinition of the NOt Equal operator.

Properties

- int ID `[get, set]`

  Property to set and get the ID of a Make.
- string Name `[get, set]`

  Property to get and set the Name of a Make.

5.8.1 Detailed Description

Purpose: Definition of Make and methods to deal with Make operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:22:09 AM.

Definition at line 23 of file Make.cs.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Make() `[1/2]`

```
trabalhoPOO_27967.Make.Make ( )
```

The default Constructor.

Definition at line 38 of file Make.cs.

5.8.2.2 Make() `[2/2]`

```
trabalhoPOO_27967.Make.Make (
            string name )
```

Constructor when the name of the make is given.

Parameters

id	
name	

Definition at line 49 of file Make.cs.

5.8.3 Member Function Documentation

5.8.3.1 Equals()

```
override bool trabalhoPOO_27967.Make.Equals (
            object obj )
```

The redefinition of the Equals() Method, to verify if a make matches another one.

Parameters

obj	

Returns

Veriffies if the object given is null.

Casts the object to be Make.

Definition at line 98 of file Make.cs.

5.8.3.2 operator"!=()

```
static bool trabalhoPOO_27967.Make.operator!= (
            Make m1,
            Make m2 )  [static]
```

The redefinition of the NOt Equal operator.

Parameters

m1	
m2	

Returns

Definition at line 128 of file Make.cs.

5.8.3.3 operator==()

```
static bool trabalhoPOO_27967.Make.operator== (
            Make m1,
            Make m2 )  [static]
```

The redefinition of the Equal operator.

Parameters

m1	
m2	

Returns

Definition at line 117 of file Make.cs.

5.8.3.4 ToString()

```
override string trabalhoPOO_27967.Make.ToString ( )
```

Override of the ToString() method to convert the data of a Make to a string.

Returns

Definition at line 88 of file Make.cs.

5.8.4 Property Documentation

5.8.4.1 ID

```
int trabalhoPOO_27967.Make.ID  [get], [set]
```

Property to set and get the ID of a Make.

Definition at line 64 of file Make.cs.

5.8.4.2 Name

```
string trabalhoPOO_27967.Make.Name  [get], [set]
```

Property to get and set the Name of a Make.

Definition at line 73 of file Make.cs.

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

- Make/Make.cs

5.9 trabalhoPOO_27967.Makes Class Reference

Purpose:This file has the definition and methods to work with the plurality of Make. Created by: Jose Alves a27967 Created on: 11/14/2024 4:33:51 PM.

Public Member Functions

- Makes ()

  The default Constructor.
- Makes (List< Make > m)

  The constructor to use when a list of Make is given.
- bool AddMake (Make m)

  Method used to add a make to a list of makes.
- bool RemoveMake (Make m)

  Method used to remove a make from a list of makes.
- bool ExistMake (int id)

  Method used to verify if a make exists on a list of makes, given its ID.
- bool ExistMake (string name)

  Method used to verify if a make exists on a list of makes, given its name.

Properties

- List< Make > MakeList `[get, set]`

  The property to get and set a list of Make.

5.9.1 Detailed Description

Purpose:This file has the definition and methods to work with the plurality of Make. Created by: Jose Alves a27967 Created on: 11/14/2024 4:33:51 PM.

Definition at line 21 of file Makes.cs.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Makes() `[1/2]`

```
trabalhoPOO_27967.Makes.Makes ( )
```

The default Constructor.

Definition at line 34 of file Makes.cs.

5.9.2.2 Makes() `[2/2]`

```
trabalhoPOO_27967.Makes.Makes (
            List< Make > m )
```

The constructor to use when a list of Make is given.

Parameters

m	

Definition at line 43 of file Makes.cs.

5.9.3 Member Function Documentation

5.9.3.1 AddMake()

```
bool trabalhoPOO_27967.Makes.AddMake (
            Make m )
```

Method used to add a make to a list of makes.

Parameters

m	

Returns

Definition at line 71 of file Makes.cs.

5.9.3.2 ExistMake() [1/2]

```
bool trabalhoPOO_27967.Makes.ExistMake (
            int id )
```

Method used to verify if a make exists on a list of makes, given its ID.

Parameters

id	

Returns

Definition at line 101 of file Makes.cs.

5.9.3.3 ExistMake() [2/2]

```
bool trabalhoPOO_27967.Makes.ExistMake (
            string name )
```

Method used to verify if a make exists on a list of makes, given its name.

Parameters

name	

Returns

Definition at line 118 of file Makes.cs.

5.9.3.4 RemoveMake()

```
bool trabalhoPOO_27967.Makes.RemoveMake (
            Make m )
```

Method used to remove a make from a list of makes.

Parameters

m	

Returns

Definition at line 83 of file Makes.cs.

5.9.4 Property Documentation

5.9.4.1 MakeList

```
List<Make> trabalhoPOO_27967.Makes.MakeList [get], [set]
```

The property to get and set a list of Make.

Definition at line 53 of file Makes.cs.

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

- Make/Makes.cs

5.10 trabalhoPOO_27967.Product Class Reference

Purpose: Definition of product and methods to deal with product operations. Created by: Jose Alves a27967 Created on: 11/2/2024 4:40:12 PM.

Public Member Functions

- Product ()

  The default Constructor.
- Product (string reff, decimal pri, Warranty war, Make ma, Category cat)

  Constructor for when the reference, price and warranty duration are given.
- override bool Equals (object obj)

  Redefinition of the method to compare two products.
- override string ToString ()

  Override of the ToString() Method to convert the data of a product into a string.

Static Public Member Functions

- static bool operator== (Product p1, Product p2)

  Redefinition of the equal operator.
- static bool operator!= (Product p1, Product p2)

  Redefinition of the different operator.

Properties

- string Reference `[get, set]`

  Property to set and get the reference of a product.
- decimal Price `[get, set]`

  Property to get and set the price of a product.
- Make Make `[get, set]`

  Property to set and get the Make of a product.
- Category Category `[get, set]`

  Property to set and get the Category of a product.
- int Stock `[get, set]`

  Property to get and set the existing stock of a product.
- Warranty Warranty `[get, set]`

5.10.1 Detailed Description

Purpose: Definition of product and methods to deal with product operations. Created by: Jose Alves a27967 Created on: 11/2/2024 4:40:12 PM.

Definition at line 25 of file Product.cs.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 Product() [1/2]

```
trabalhoPOO_27967.Product.Product ( )
```

The default Constructor.

Definition at line 43 of file Product.cs.

5.10.2.2 Product() [2/2]

```
trabalhoPOO_27967.Product.Product (
             string reff,
             decimal pri,
             Warranty war,
             Make ma,
             Category cat )
```

Constructor for when the reference, price and warranty duration are given.

Parameters

reff	
pri	

Definition at line 57 of file Product.cs.

5.10.3 Member Function Documentation

5.10.3.1 Equals()

```
override bool trabalhoPOO_27967.Product.Equals (
             object obj )
```

Redefinition of the method to compare two products.

Parameters

obj	

Returns

Definition at line 133 of file Product.cs.

5.10.3.2 operator"!=()

```
static bool trabalhoPOO_27967.Product.operator!= (
             Product p1,
             Product p2 )  [static]
```

Redefinition of the different operator.

Parameters

p1	
p2	

Returns

Definition at line 158 of file Product.cs.

5.10.3.3 operator==()

```
static bool trabalhoPOO_27967.Product.operator== (
            Product p1,
            Product p2 )  [static]
```

Redefinition of the equal operator.

Parameters

p1	
p2	

Returns

Definition at line 147 of file Product.cs.

5.10.3.4 ToString()

```
override string trabalhoPOO_27967.Product.ToString ( )
```

Override of the ToString() Method to convert the data of a product into a string.

Returns

Definition at line 167 of file Product.cs.

5.10.4 Property Documentation

5.10.4.1 Category

```
Category trabalhoPOO_27967.Product.Category  [get], [set]
```

Property to set and get the Category of a product.

Definition at line 103 of file Product.cs.

5.10.4.2 Make

```
Make trabalhoPOO_27967.Product.Make  [get], [set]
```

Property to set and get the Make of a product.

Definition at line 93 of file Product.cs.

5.10.4.3 Price

```
decimal trabalhoPOO_27967.Product.Price  [get], [set]
```

Property to get and set the price of a product.

Definition at line 84 of file Product.cs.

5.10.4.4 Reference

```
string trabalhoPOO_27967.Product.Reference  [get], [set]
```

Property to set and get the reference of a product.

Definition at line 74 of file Product.cs.

5.10.4.5 Stock

```
int trabalhoPOO_27967.Product.Stock  [get], [set]
```

Property to get and set the existing stock of a product.

Definition at line 112 of file Product.cs.

5.10.4.6 Warranty

```
Warranty trabalhoPOO_27967.Product.Warranty  [get], [set]
```

Definition at line 118 of file Product.cs.

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

- Product/Product.cs

5.11 trabalhoPOO_27967.Products Class Reference

Purpose: Class to manage a group of more than one product. Created by: Jose Alves a27967 Created on↩ : 11/9/2024 6:34:19 PM.

Public Member Functions

- Products ()

  The default Constructor.
- Products (List< Product > products)

  The constructor to use when list of Product is given.
- override string ToString ()

  Override of the ToString() Method to convert the data of a list fo products to a string.
- decimal ValueInPosition (int p)

  This method returns the price of a product, given a certain array position.
- bool InsertProduct (Product p)

  This method inserts a product in a list of products.
- bool ExistProduct (string reff)

  Method used to verify if a product is on a products' list.
- bool RemoveProduct (Product p)

  Method used to remove a product from a Products' list.
- Product SearchProduct (string reff)

  This method searches for a product in an array, given its refference.
- decimal TotalPrice ()

  Method used to calculate the total price of products in a list of products.
- DateTime WarratyExpirationDateForProduct (DateTime date, string reff)

  Method used to calculate the warranty's expiration date of a product on a list of products.

Properties

- List< Product > Prods `[get, set]`

  Property used to get and set the list of products.

5.11.1 Detailed Description

Purpose: Class to manage a group of more than one product. Created by: Jose Alves a27967 Created on↩ : 11/9/2024 6:34:19 PM.

Definition at line 25 of file Products.cs.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 Products() [1/2]

```
trabalhoPOO_27967.Products.Products ( )
```

The default Constructor.

Definition at line 38 of file Products.cs.

5.11.2.2 Products() [2/2]

```
trabalhoPOO_27967.Products.Products (
            List< Product > products )
```

The constructor to use when list of Product is given.

Parameters

products	

Definition at line 47 of file Products.cs.

5.11.3 Member Function Documentation

5.11.3.1 ExistProduct()

```
bool trabalhoPOO_27967.Products.ExistProduct (
            string reff )
```

Method used to verify if a product is on a products' list.

Parameters

reff	

Returns

Definition at line 115 of file Products.cs.

5.11.3.2 InsertProduct()

```
bool trabalhoPOO_27967.Products.InsertProduct (
            Product p )
```

This method inserts a product in a list of products.

Parameters

p	

Returns

Returns true or False, depending on whether or not it succeeded in inserting the product into the list.

Definition at line 103 of file Products.cs.

5.11.3.3 RemoveProduct()

```
bool trabalhoPOO_27967.Products.RemoveProduct (
            Product p )
```

Method used to remove a product from a Products' list.

Parameters

p	

Returns

Product removed successfully

Product was not removed.

Definition at line 129 of file Products.cs.

5.11.3.4 SearchProduct()

```
Product trabalhoPOO_27967.Products.SearchProduct (
             string reff )
```

This method searches for a product in an array, given its refference.

Parameters

reff	

Returns

Returns the product if found

Definition at line 145 of file Products.cs.

5.11.3.5 ToString()

```
override string trabalhoPOO_27967.Products.ToString ( )
```

Override of the ToString() Method to convert the data of a list fo products to a string.

Returns

Definition at line 75 of file Products.cs.

5.11.3.6 TotalPrice()

```
decimal trabalhoPOO_27967.Products.TotalPrice ( )
```

Method used to calculate the total price of products in a list of products.

Returns

Definition at line 158 of file Products.cs.

5.11.3.7 ValueInPosition()

```
decimal trabalhoPOO_27967.Products.ValueInPosition (
            int p )
```

This method returns the price of a product, given a certain array position.

Parameters

p	Position in array.

Returns

Definition at line 93 of file Products.cs.

5.11.3.8 WarratyExpirationDateForProduct()

```
DateTime trabalhoPOO_27967.Products.WarratyExpirationDateForProduct (
            DateTime date,
            string reff )
```

Method used to calculate the warranty's expiration date of a product on a list of products.

Parameters

date	
reff	

Returns

Definition at line 170 of file Products.cs.

5.11.4 Property Documentation

5.11.4.1 Prods

```
List<Product> trabalhoPOO_27967.Products.Prods  [get], [set]
```

Property used to get and set the list of products.

Definition at line 60 of file Products.cs.

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

- Product/Products.cs

5.12 trabalhoPOO_27967.Sale Class Reference

Purpose: Definition of Sale and methods to deal with Sale operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:53 AM.

Public Member Functions

- Sale ()

  The default Constructor.
- Sale (Client client, Products products, Campaign camp)

  Constructor used when a client, a product array and a campaign to be used are given.
- override bool Equals (object obj)

  Redefinition of the Equals method to compare two sales.
- override string ToString ()
- decimal TotalPrice ()

  Method to calculate the total price of a sale, given the products list and a campaign code.
- bool InsertProductOnSale (Product p)

  Method used to insert a product on a sale's list.
- bool RemoveProductFromSale (Product p)

  Method used to remove a product from a sale.
- bool ExistProductOnSale (Product p)

  Method used to verify if a product is on a sale.
- DateTime WarrantyExpirationDate (string reff)

  Method to calculate when a warranty is due to expire.

Static Public Member Functions

- static bool operator== (Sale s1, Sale s2)

  Redefiniiton of the equal operator.
- static bool operator!= (Sale s1, Sale s2)

  Redefinition of the different operator.

Properties

- int Id `[get, set]`

  Property used to get and set the ID of a Sale.
- Client Client `[get, set]`

  Property used to get and set the information of the Client who made the purchase.
- Products Products `[get, set]`

  Property used to get and set the list of products in a Sale.
- decimal TotPrice `[get, set]`

  Property used to get and set the total price of a Sale.
- DateTime SaleDate `[get, set]`

  Property used to get and set the Date of a Sale.
- Campaign Campaigns `[get, set]`

  Property used to get and set the Campaigns applicable to a Sale.

5.12.1 Detailed Description

Purpose: Definition of Sale and methods to deal with Sale operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:53 AM.

Definition at line 25 of file Sale.cs.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Sale() [1/2]

```
trabalhoPOO_27967.Sale.Sale ( )
```

The default Constructor.

Definition at line 45 of file Sale.cs.

5.12.2.2 Sale() [2/2]

```
trabalhoPOO_27967.Sale.Sale (
            Client client,
            Products products,
            Campaign camp )
```

Constructor used when a client, a product array and a campaign to be used are given.

Parameters

client	
products	
camp	

Definition at line 58 of file Sale.cs.

5.12.3 Member Function Documentation

5.12.3.1 Equals()

```
override bool trabalhoPOO_27967.Sale.Equals (
            object obj )
```

Redefinition of the Equals method to compare two sales.

Parameters

obj	

Returns

Definition at line 135 of file Sale.cs.

5.12.3.2 ExistProductOnSale()

```
bool trabalhoPOO_27967.Sale.ExistProductOnSale (
             Product p )
```

Method used to verify if a product is on a sale.

Parameters

p	

Returns

Definition at line 220 of file Sale.cs.

5.12.3.3 InsertProductOnSale()

```
bool trabalhoPOO_27967.Sale.InsertProductOnSale (
             Product p )
```

Method used to insert a product on a sale's list.

Parameters

p	

Returns

Definition at line 200 of file Sale.cs.

5.12.3.4 operator"!=()

```
static bool trabalhoPOO_27967.Sale.operator!= (
             Sale s1,
             Sale s2 )  [static]
```

Redefinition of the different operator.

Parameters

s1	
s2	

Returns

Definition at line 158 of file Sale.cs.

5.12.3.5 operator==()

```
static bool trabalhoPOO_27967.Sale.operator== (
            Sale s1,
            Sale s2 )  [static]
```

Redefiniiton of the equal operator.

Parameters

s1	
s2	

Returns

Definition at line 147 of file Sale.cs.

5.12.3.6 RemoveProductFromSale()

```
bool trabalhoPOO_27967.Sale.RemoveProductFromSale (
            Product p )
```

Method used to remove a product from a sale.

Parameters

p	

Returns

Definition at line 210 of file Sale.cs.

5.12.3.7 ToString()

```
override string trabalhoPOO_27967.Sale.ToString ( )
```

\u20AC é o unicode para o símbolo de euro.

Definition at line 163 of file Sale.cs.

5.12.3.8 TotalPrice()

```
decimal trabalhoPOO_27967.Sale.TotalPrice ( )
```

Method to calculate the total price of a sale, given the products list and a campaign code.

Parameters

products	
camp	

Returns

Definition at line 181 of file Sale.cs.

5.12.3.9 WarrantyExpirationDate()

```
DateTime trabalhoPOO_27967.Sale.WarrantyExpirationDate (
             string reff )
```

Method to calculate when a warranty is due to expire.

Parameters

s	
reff	

Returns

Definition at line 231 of file Sale.cs.

5.12.4 Property Documentation

5.12.4.1 Campaigns

```
Campaign trabalhoPOO_27967.Sale.Campaigns  [get], [set]
```

Property used to get and set the Campaigns applicable to a Sale.

Definition at line 120 of file Sale.cs.

5.12.4.2 Client

```
Client trabalhoPOO_27967.Sale.Client  [get], [set]
```

Property used to get and set the information of the Client who made the purchase.

Definition at line 84 of file Sale.cs.

5.12.4.3 Id

```
int trabalhoPOO_27967.Sale.Id  [get], [set]
```

Property used to get and set the ID of a Sale.

Definition at line 75 of file Sale.cs.

5.12.4.4 Products

```
Products trabalhoPOO_27967.Sale.Products  [get], [set]
```

Property used to get and set the list of products in a Sale.

Definition at line 93 of file Sale.cs.

5.12.4.5 SaleDate

```
DateTime trabalhoPOO_27967.Sale.SaleDate  [get], [set]
```

Property used to get and set the Date of a Sale.

Definition at line 111 of file Sale.cs.

5.12.4.6 TotPrice

```
decimal trabalhoPOO_27967.Sale.TotPrice  [get], [set]
```

Property used to get and set the total price of a Sale.

Definition at line 102 of file Sale.cs.

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

- Sale/Sale.cs

5.13 trabalhoPOO_27967.Sales Class Reference

Purpose: Class with the agregation of sales of a store. Created by: Jose Alves a27967 Created on: 11/10/2024 7:42:03 PM.

Public Member Functions

- Sales ()

  The default Constructor.
- Sales (List< Sale > sales)

  The constructor to use when the sales' List is given.
- Sale GetSale (int idSale)

  Method to find a sale in a list of sales, given its ID.
- bool AddSale (Sale sale)

  Method used to add a sale to a sales' list.

Properties

- List< Sale > SalesStored `[get, set]`

  Property used to get and set a list of sales.

5.13.1 Detailed Description

Purpose: Class with the agregation of sales of a store. Created by: Jose Alves a27967 Created on: 11/10/2024 7:42:03 PM.

Definition at line 21 of file Sales.cs.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 Sales() `[1/2]`

```
trabalhoPOO_27967.Sales.Sales ( )
```

The default Constructor.

Definition at line 34 of file Sales.cs.

5.13.2.2 Sales() `[2/2]`

```
trabalhoPOO_27967.Sales.Sales (
            List< Sale > sales )
```

The constructor to use when the sales' List is given.

Parameters

sales	

Definition at line 43 of file Sales.cs.

5.13.3 Member Function Documentation

5.13.3.1 AddSale()

```
bool trabalhoPOO_27967.Sales.AddSale (
             Sale sale )
```

Method used to add a sale to a sales' list.

Parameters

sale	

Returns

Definition at line 88 of file Sales.cs.

5.13.3.2 GetSale()

```
Sale trabalhoPOO_27967.Sales.GetSale (
             int idSale )
```

Method to find a sale in a list of sales, given its ID.

Parameters

idSale	

Returns

Definition at line 74 of file Sales.cs.

5.13.4 Property Documentation

5.13.4.1 SalesStored

```
List<Sale> trabalhoPOO_27967.Sales.SalesStored  [get], [set]
```

Property used to get and set a list of sales.

Definition at line 55 of file Sales.cs.

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

- Sale/Sales.cs

5.14 trabalhoPOO_27967.Store.Store Class Reference

Purpose: This class has the definition and properties to manage a store. Created by: Jose Alves a27967 Created on: 11/14/2024 5:01:23 PM.

Public Member Functions

- Store ()
  The default Constructor.
- Store (Clients cl, Products p, Sales s, Makes m, Categories c, Warranties w)
  The constructor to use when all the lists are given.

Properties

- Clients ClientLIst `[get, set]`
  The property used to get and set the clients' list.
- Products ProdList `[get, set]`
  The property to get and set de products' list.
- Sales SaleList `[get, set]`
  The property to get and set de sales' list.
- Makes MakeList `[get, set]`
  The property to get and set de makes' list.
- Categories CatList `[get, set]`
  The property to get and set de categories' list.
- Warranties WarrantList `[get, set]`
  The property to get and set de warranties' list.

5.14.1 Detailed Description

Purpose: This class has the definition and properties to manage a store. Created by: Jose Alves a27967 Created on: 11/14/2024 5:01:23 PM.

Definition at line 20 of file Store.cs.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 Store() [1/2]

```
trabalhoPOO_27967.Store.Store.Store ( )
```

The default Constructor.

Definition at line 38 of file Store.cs.

5.14.2.2 Store() [2/2]

```
trabalhoPOO_27967.Store.Store.Store (
             Clients cl,
             Products p,
             Sales s,
             Makes m,
             Categories c,
             Warranties w )
```

The constructor to use when all the lists are given.

Parameters

cl	
p	
s	
m	
c	
w	

Definition at line 57 of file Store.cs.

5.14.3 Property Documentation

5.14.3.1 CatList

```
Categories trabalhoPOO_27967.Store.Store.CatList  [get], [set]
```

The property to get and set de categories' list.

Definition at line 109 of file Store.cs.

5.14.3.2 ClientLIst

```
Clients trabalhoPOO_27967.Store.Store.ClientLIst  [get], [set]
```

The property used to get and set the clients' list.

Definition at line 73 of file Store.cs.

5.14.3.3 MakeList

```
Makes trabalhoPOO_27967.Store.Store.MakeList  [get], [set]
```

The property to get and set de makes' list.

Definition at line 100 of file Store.cs.

5.14.3.4 ProdList

```
Products trabalhoPOO_27967.Store.Store.ProdList  [get], [set]
```

The property to get and set de products' list.

Definition at line 82 of file Store.cs.

5.14.3.5 SaleList

```
Sales trabalhoPOO_27967.Store.Store.SaleList  [get], [set]
```

The property to get and set de sales' list.

Definition at line 91 of file Store.cs.

5.14.3.6 WarrantList

```
Warranties trabalhoPOO_27967.Store.Store.WarrantList  [get], [set]
```

The property to get and set de warranties' list.

Definition at line 118 of file Store.cs.

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

- Store/Store.cs

5.15 trabalhoPOO_27967.Warranties Class Reference

Purpose:This file has the definition and methods to work with the plurality of Warranty. Created by: Jose Alves a27967 Created on: 11/14/2024 4:20:11 PM.

Public Member Functions

- Warranties ()

  The default Constructor.
- Warranties (List< Warranty > warrants)

  The constructor to use when a list of Warranties is given.
- bool AddWarranty (Warranty w)
- bool RemoveWarranty (Warranty w)

  Method used to remove a warranty from a list of warranties.
- bool ExistWarranty (string id)

  Method used to confirm if a campaign exists on a list of campaigns.

Properties

- List< Warranty > Warrants `[get, set]`

  Property used to get and set the list of warranties.

5.15.1 Detailed Description

Purpose:This file has the definition and methods to work with the plurality of Warranty. Created by: Jose Alves a27967 Created on: 11/14/2024 4:20:11 PM.

Definition at line 21 of file Warranties.cs.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 Warranties() [1/2]

```
trabalhoPOO_27967.Warranties.Warranties ( )
```

The default Constructor.

Definition at line 34 of file Warranties.cs.

5.15.2.2 Warranties() [2/2]

```
trabalhoPOO_27967.Warranties.Warranties (
            List< Warranty > warrants )
```

The constructor to use when a list of Warranties is given.

Parameters

warrants	

Definition at line 43 of file Warranties.cs.

5.15.3 Member Function Documentation

5.15.3.1 AddWarranty()

```
bool trabalhoPOO_27967.Warranties.AddWarranty (
            Warranty w )
```

Method used to add a warranty to a list of warranties.

Parameters

c	

Returns

Definition at line 74 of file Warranties.cs.

5.15.3.2 ExistWarranty()

```
bool trabalhoPOO_27967.Warranties.ExistWarranty (
            string id )
```

Method used to confirm if a campaign exists on a list of campaigns.

Parameters

id	

Returns

Definition at line 104 of file Warranties.cs.

5.15.3.3 RemoveWarranty()

```
bool trabalhoPOO_27967.Warranties.RemoveWarranty (
            Warranty w )
```

Method used to remove a warranty from a list of warranties.

Parameters

camp	

Returns

Definition at line 86 of file Warranties.cs.

5.15.4 Property Documentation

5.15.4.1 Warrants

```
List<Warranty> trabalhoPOO_27967.Warranties.Warrants  [get], [set]
```

Property used to get and set the list of warranties.

Definition at line 56 of file Warranties.cs.

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

- Warranty/Warranties.cs

5.16 trabalhoPOO_27967.Warranty Class Reference

Purpose: This class contains the definition and methods to manage warranties. Created by: Jose Alves a27967 Created on: 11/13/2024 4:17:18 PM.

Public Member Functions

- Warranty ()

  The default Constructor.
- Warranty (string prodID, int durationInYears, string conditions)

  The constructor to use when the ID of the product, duration (in years) and the conditions are given.
- override string ToString ()

  Method to show the information of a Warranty.
- override bool Equals (object obj)

  Redefine the Equals operator to verify if a warranty matches the other.
- DateTime ExpirationDate (Sale s, string reff)

  Method to calculate when a warranty is due to expire.

Static Public Member Functions

- static bool operator== (Warranty w1, Warranty w2)

  Redefinition of the == operator.
- static bool operator!= (Warranty w1, Warranty w2)

  Redefinition of the != operator.

Properties

- string ProdID `[get, set]`

  Property to get and set the ID of the product to which the warranty is about.
- int DurationInYears `[get, set]`

  The property to set and get the duration of a warranty (in years).
- string Conditions `[get, set]`

  The property to get and set the terms of a warranty.

5.16.1 Detailed Description

Purpose: This class contains the definition and methods to manage warranties. Created by: Jose Alves a27967 Created on: 11/13/2024 4:17:18 PM.

Definition at line 23 of file Warranty.cs.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Warranty() [1/2]

```
trabalhoPOO_27967.Warranty.Warranty ( )
```

The default Constructor.

Definition at line 38 of file Warranty.cs.

5.16.2.2 Warranty() [2/2]

```
trabalhoPOO_27967.Warranty.Warranty (
            string prodID,
            int durationInYears,
            string conditions )
```

The constructor to use when the ID of the product, duration (in years) and the conditions are given.

Parameters

prodID	
durationInYears	
conditions	

Definition at line 51 of file Warranty.cs.

5.16.3 Member Function Documentation

5.16.3.1 Equals()

```
override bool trabalhoPOO_27967.Warranty.Equals (
            object obj )
```

Redefine the Equals operator to verify if a warranty matches the other.

Parameters

obj	

Returns

Veriffies if the object given is null.

Casts the object to be Warranty.

Definition at line 114 of file Warranty.cs.

5.16.3.2 ExpirationDate()

```
DateTime trabalhoPOO_27967.Warranty.ExpirationDate (
            Sale s,
            string reff )
```

Method to calculate when a warranty is due to expire.

Parameters

s	
reff	

Returns

Definition at line 157 of file Warranty.cs.

5.16.3.3 operator"!=()

```
static bool trabalhoPOO_27967.Warranty.operator!= (
            Warranty w1,
            Warranty w2 )  [static]
```

Redefinition of the != operator.

Parameters

w1	
w2	

Returns

Definition at line 144 of file Warranty.cs.

5.16.3.4 operator==()

```
static bool trabalhoPOO_27967.Warranty.operator== (
            Warranty w1,
            Warranty w2 )  [static]
```

Redefinition of the == operator.

Parameters

w1	
w2	

Returns

Definition at line 133 of file Warranty.cs.

5.16.3.5 ToString()

```
override string trabalhoPOO_27967.Warranty.ToString ( )
```

Method to show the information of a Warranty.

Returns

Definition at line 99 of file Warranty.cs.

5.16.4 Property Documentation

5.16.4.1 Conditions

```
string trabalhoPOO_27967.Warranty.Conditions  [get], [set]
```

The property to get and set the terms of a warranty.

Definition at line 85 of file Warranty.cs.

5.16.4.2 DurationInYears

```
int trabalhoPOO_27967.Warranty.DurationInYears  [get], [set]
```

The property to set and get the duration of a warranty (in years).

Definition at line 76 of file Warranty.cs.

5.16.4.3 ProdID

```
string trabalhoPOO_27967.Warranty.ProdID  [get], [set]
```

Property to get and set the ID of the product to which the warranty is about.

Definition at line 67 of file Warranty.cs.

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

- Warranty/Warranty.cs

Chapter 6

File Documentation

6.1 BestSale.cs File Reference

Classes

- class trabalhoPOO_27967.BestSale

Namespaces

- namespace trabalhoPOO_27967

6.2 BestSale.cs

Go to the documentation of this file.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions; //Used to verify if a string meets certain criteria.

namespace trabalhoPOO_27967
{
    class BestSale
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = System.Text.Encoding.UTF8;

            Client clientTest = new Client("Jose Alves","962310421");
            Client clientTest2 = new Client("Rui Jordao", "931250420");
            Client clientTest1 = new Client();

            Make benfica = new Make("Benfica");
            Category futebol = new Category("Futebol");
            Product product1 = new Product("1A34", 23.9m, new Warranty("1A34",3, "Nao gostar de batatas."), benfica, futebol );

            Make braga = new Make("Braga");
            Product product2 = new Product("2V45", 15.9m, new Warranty("2V45", 3, "Nao gostar de peixe."), braga, futebol);

            Sale sale = new Sale();
            sale.InsertProductOnSale(product1);
            sale.InsertProductOnSale(product2);

            Console.WriteLine(sale.ToString());

            Clients clientes = new Clients();
            clientes.AddClient(clientTest);
            clientes.AddClient(clientTest2);

            foreach (Client client in clientes.ClientList)
            {
                Console.WriteLine(client.ToString());
            }



        }
    }
}
```

6.3 Campaign/Campaign.cs File Reference

Classes

- class trabalhoPOO_27967.Campaign

  Purpose: Definition of Campaign and methods to deal with Campaign operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:43 AM.

Namespaces

- namespace trabalhoPOO_27967

6.4 Campaign.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/6/2024 11:21:43 AM</date>
*   <description></description>
**/
using System;

namespace trabalhoPOO_27967
{
    public class Campaign
    {
        #region Attributes
        int _id;
        string _name;
        decimal _discount;
        DateTime _startDate;
        DateTime _endDate;
        static int _campaignCount;
        #endregion

        #region Methods

        #region Constructors

        public Campaign()
        {
            _id = ++_campaignCount;
            _name = string.Empty;
            _discount = 0m;
            _startDate = DateTime.MinValue;
            _endDate = DateTime.MaxValue;
        }

        public Campaign(string name, decimal discount, DateTime startDate, DateTime endDate)
        {
            _id = ++_campaignCount;
            _name = name;
            _discount = discount;
            _startDate = startDate;
            _endDate = endDate;
        }





        #endregion

        #region Properties

        public int Id{
            get { return _id; }
            set { _id = value; }
        }

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }

        public decimal Discount
        {
            get { return _discount; }
            set { _discount = value; }
        }

        public DateTime StartDate
        {
            get { return _startDate; }
            set { _startDate = value; }
        }

        public DateTime EndDate
        {
            get { return _endDate; }
            set { _endDate = value; }
        }

        public int CampaignCount
        {
            get { return _campaignCount; }
            set { _campaignCount = value; }
        }
        #endregion



        #region Overrides
        public override bool Equals(object obj)
        {
            if (obj == null)
            {
                return false;
            }

            Campaign camp = obj as Campaign;
            return (this.Id == camp.Id && _name == camp.Name);
        }

        public static bool operator ==(Campaign camp1, Campaign camp2)
        {
            return (camp1.Equals(camp2));
        }

        public static bool operator !=(Campaign camp1, Campaign camp2)
        {
            return !(camp1.Equals(camp2));
        }
        #endregion

        #region OtherMethods

        static public bool VerifyApplicability(Campaign camp)
        {
            if (camp.StartDate <= DateTime.Now && camp.EndDate >= DateTime.Now)
            {
                return true;
            }

            return false;
        }
        #endregion

        #region Destructor
        ~Campaign()
        {
        }
        #endregion

        #endregion
    }
}
```

6.5 Campaign/Campaigns.cs File Reference

Classes

- class trabalhoPOO_27967.Campaigns

  Purpose:This file has the definition and methods to work with the plurality of Campaign. Created by: Jose Alves a27967 Created on: 11/14/2024 3:57:04 PM.

Namespaces

- namespace trabalhoPOO_27967

6.6 Campaigns.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/14/2024 3:57:04 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;
using System.Xml.Linq;

namespace trabalhoPOO_27967
{
    public class Campaigns
    {
        #region Attributes
        List<Campaign> _camps;
        #endregion

        #region Methods

        #region Constructors

        public Campaigns()
        {
            _camps = new List<Campaign>();
        }

        public Campaigns(List<Campaign> p) {
            _camps = p;
        }

        #endregion

        #region Properties
        public List<Campaign> Camps
        {
            get{  return _camps; }
            set{ _camps = value; }
        }
        #endregion



        #region Overrides

        #endregion

        #region OtherMethods
        public bool AddCampaign(Campaign c)
        {
            if(c==null)return false;
            _camps.Add(c);
            return true;
        }

        public bool RemoveCampaign(Campaign camp)
        {
            foreach (Campaign c in _camps)
            {
                if (c == camp)
                {
                    _camps.Remove(camp);
                    return true;
                }
            }
            return false;
        }

        public bool ExistCampaign(int id)
        {
            foreach (Campaign c in _camps)
            {
                if (c.Id == id)
                {
                    return true;
                }
            }
            return false;
        }
        #endregion

        #region Destructor
        ~Campaigns()
        {
        }
        #endregion

        #endregion
    }
}
```

6.7 Category/Categories.cs File Reference

Classes

- class trabalhoPOO_27967.Categories

  Purpose:This file has the definition and methods to work with the plurality of Category. Created by: Jose Alves a27967 Created on: 11/14/2024 4:45:58 PM.

Namespaces

- namespace trabalhoPOO_27967

6.8 Categories.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.Category.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/14/2024 4:45:58 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;

namespace trabalhoPOO_27967
{
    public class Categories
    {
        #region Attributes
        List<Category> _cats;
        #endregion

        #region Methods

        #region Constructors

        public Categories()
        {
            _cats = new List<Category>();
        }

        public Categories(List<Category> cats)
        {
            _cats = cats;
        }


        #endregion

        #region Properties
        public List<Category> Cats
        {
            get { return _cats; }
            set { _cats = value; }
        }
        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        public bool AddCategory(Category c)
        {
            if (c == null) return false;
            _cats.Add(c);
            return true;
        }

        public bool RemoveCategory(Category c)
        {
            foreach (Category cate in _cats)
            {
                if (cate == c)
                {
                    _cats.Remove(cate);
                    return true;
                }
            }
            return false;
        }

        public bool ExistCategory(int id)
        {
            foreach (Category cate in _cats)
            {
                if (cate.Id == id)
                {
                    return true;
                }
            }
            return false;
        }

        public bool ExistCategory(string name)
        {
            foreach (Category cate in _cats)
            {
                if (cate.Name == name)
                {
                    return true;
                }
            }
            return false;
        }

        #endregion

        #region Destructor
        ~Categories()
        {
        }
        #endregion

        #endregion
    }
}
```

6.9 Category/Category.cs File Reference

Classes

- class trabalhoPOO_27967.Category

  Purpose: Definition of Category and methods to deal with Category operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:20:47 AM.

Namespaces

- namespace trabalhoPOO_27967

6.10 Category.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/6/2024 11:20:47 AM</date>
*   <description></description>
**/
using System;

namespace trabalhoPOO_27967
{
    public class Category
    {
        #region Attributes
        int _id;
        string _name;
        static int _catCount=0;
        #endregion

        #region Methods

        #region Constructors

        public Category()
        {
            _id = ++_catCount;
            _name = string.Empty;
        }

        public Category(string name)
        {
            _id = ++_catCount;
            _name = name;
        }
        #endregion

        #region Properties

        public int Id
        {
            get { return _id; }
            set { _id = value; }
        }

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }
        #endregion



        #region Overrides
        public override bool Equals(object obj)
        {
            if (obj == null)
            {
                return false;
            }

            Category cat = obj as Category;
            return (this.Id == cat.Id || _name == cat.Name);
        }

        public static bool operator ==(Category cat1, Category cat2)
        {
            return (cat1.Equals(cat2));
        }

        public static bool operator !=(Category cat1, Category cat2)
        {
            return !(cat1.Equals(cat2));
        }
        #endregion

        #region OtherMethods
        #endregion

        #region Destructor
        ~Category()
        {
        }
        #endregion

        #endregion
    }
}
```

6.11 Client/Client.cs File Reference

Classes

- class trabalhoPOO_27967.Client

  Purpose: Definition of Client and methods to deal with Client operations. Created by: Jose Alves a27967 Created on: 10/29/2024 4:23:56 PM.

Namespaces

- namespace trabalhoPOO_27967

6.12 Client.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>10/29/2024 4:23:56 PM</date>
*   <description></description>
**/
using System;
using System.Text;
using System.Text.RegularExpressions;
using static System.Net.Mime.MediaTypeNames;

namespace trabalhoPOO_27967
{
    public class Client
    {
        #region Attributes
        int _clientID;
        string _name;
        string _contact;
        static int _clientCount=0;
        #endregion

        #region Methods

        #region Constructors

        public Client()
        {
            _clientID = ++_clientCount ;
            _name = "No Name";
            _contact = "999999999";
        }

        public Client(string n, string c)
        {
            _clientID = ++_clientCount;
            _name = n;
            _contact = c;
            //COMO POSSO FAZER PARA TESTAR SE A STRING PODE SER CONTATCO? DEVO FAZE-LO NO CONSTRUTOR OU FORA?
        }

        #endregion

        #region Properties
        public int ClientID
        {
            get { return _clientID; }
            set { _clientID = value; }
        }

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }

        public string Contact
        {
            get { return _contact; }
            set
            {
                string pattern = @"^(2|9)\d{8}$"; //Defines the pattern to be a number starting by 9 or 2 with 8 more numbers after (as a portuguese mobile or landline number).
                bool isGood = Regex.IsMatch(value, pattern); //Verifies if the value meets the criteria.

                if (isGood) _contact = value;

                //COMO POSSO RETORNAR UM ERRO CASO A STRING NAO CORRESPONDA?
            }
        }

        public static int ClientCount
        {
            get { return _clientCount; }

            set { _clientCount = value; }
        }
        #endregion



        #region Overrides

        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            sb.AppendLine($"Client ID: {_clientID}");
            sb.AppendLine($"Name: {_name}");
            sb.AppendLine($"Contact: {_contact}");

            return sb.ToString();
        }

        public override bool Equals(object obj)
        {
            if (obj == null)
            {
                return false;
            }

            Client client = obj as Client;
            return (this._clientID == client._clientID && _name == client._name);
        }

        public static bool operator == (Client cli1, Client cli2)
        {
            return( cli1.Equals(cli2) );
        }

        public static bool operator !=(Client cli1, Client cli2)
        {
            return !(cli1.Equals(cli2));
        }
        #endregion


        #region OtherMethods
        #endregion

        #region Destructor
        ~Client()
        {
        }
        #endregion

        #endregion
    }
}
```

6.13 Client/Clients.cs File Reference

Classes

- class trabalhoPOO_27967.Clients

  Purpose: Class with the definition and methods to manage a list of clients. Created by: Jose Alves a27967 Created on: 11/12/2024 9:25:28 PM.

Namespaces

- namespace trabalhoPOO_27967

6.14 Clients.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/12/2024 9:25:28 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;

```

```
namespace trabalhoPOO_27967
{
    public class Clients
    {
        #region Attributes
        static List<Client> _clientList;
        #endregion

        #region Methods

        #region Constructors

        public Clients()
        {
            _clientList = new List<Client>();
        }


        #endregion

        #region Properties

        public List<Client> ClientList
        {
            get{ return _clientList; }
            set{ _clientList = value; }
        }
        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        public bool AddClient(Client cli)
        {
            if (cli == null) return false;
            this.ClientList.Add(cli);
            return true;
        }

        public bool RemoveClient(Client cli)
        {
            foreach(Client client in _clientList)
            {
                if (client == cli)
                {
                    _clientList.Remove(cli);
                    return true;
                }
            }
            return false;
        }

        public bool ExistClient(int id)
        {
            foreach (Client client in _clientList)
            {
                if (client.ClientID == id)
                {
                    return true;
                }
            }
            return false;
        }

        public Client GetClient(int id)
        {
            foreach (Client client in _clientList)
            {
                if (client.ClientID == id)
                {
                    return client;
                }
            }
            return null;
        }
        #endregion

        #region Destructor
        ~Clients()
        {
        }
        #endregion

        #endregion
    }
}
```

6.15 Make/Make.cs File Reference

Classes

- class trabalhoPOO_27967.Make

  Purpose: Definition of Make and methods to deal with Make operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:22:09 AM.

Namespaces

- namespace trabalhoPOO_27967

6.16 Make.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/6/2024 11:22:09 AM</date>
*   <description></description>
**/
using System;
using System.Diagnostics.Contracts;
using System.Runtime.Remoting.Messaging;
using System.Text;

namespace trabalhoPOO_27967
{
    public class Make
    {
        #region Attributes
        int _id;
        string _name;
        static int _makeCount=0;
        #endregion

        #region Methods

        #region Constructors

        public Make()
        {
            _id = ++_makeCount;
            _name = string.Empty;
        }

        public Make(string name)
        {
            _id = ++_makeCount;
            _name = name;
        }



        #endregion

        #region Properties

        public int ID
        {
            get { return _id; }
            set { _id = value; }
        }

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }

        #endregion



        #region Overrides
        public override string ToString()
        {
            return ("Make : " + _name);
        }

        public override bool Equals(object obj)
        {
            if (obj == null)
            {
                return false;
            }

            Make make = obj as Make;
            return (this.ID == make.ID || this.Name == make.Name);
        }

        public static bool operator ==(Make m1, Make m2)
        {
            return (m1.Equals(m2));
        }

        public static bool operator !=(Make m1, Make m2)
        {
            return !(m1.Equals(m2));
        }
        #endregion

        #region OtherMethods
        #endregion

        #region Destructor
        ~Make()
        {
        }
        #endregion

        #endregion
    }
}
```

6.17 Make/Makes.cs File Reference

Classes

- class trabalhoPOO_27967.Makes

  Purpose:This file has the definition and methods to work with the plurality of Make. Created by: Jose Alves a27967 Created on: 11/14/2024 4:33:51 PM.

Namespaces

- namespace trabalhoPOO_27967

6.18 Makes.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/14/2024 4:33:51 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;

namespace trabalhoPOO_27967
{
    public class Makes
    {
        #region Attributes
        List<Make> _makeList;
        #endregion

        #region Methods

        #region Constructors

        public Makes()
        {
            _makeList = new List<Make>();
        }

        public Makes(List<Make> m)
        {
            _makeList = m;
        }
        #endregion

        #region Properties
        public List<Make> MakeList
        {
            get { return _makeList; }
            set { _makeList = value; }
        }
        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        public bool AddMake(Make m)
        {
            if (m == null) return false;
            _makeList.Add(m);
            return true;
        }

        public bool RemoveMake(Make m)
        {
            foreach (Make make in _makeList)
            {
                if (make == m)
                {
                    _makeList.Remove(make);
                    return true;
                }
            }
            return false;
        }

        public bool ExistMake(int id)
        {
            foreach (Make make in _makeList)
            {
                if(make.ID == id)
                {
                    return true;
                }
            }
            return false;
        }

        public bool ExistMake(string name)
        {
            foreach (Make make in _makeList)
            {
                if (make.Name == name)
                {
                    return true;
                }
            }
            return false;
        }
        #endregion

        #region Destructor
        ~Makes()
        {
        }
        #endregion

        #endregion
    }
}
```

6.19 obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs File Reference

6.20 .NETFramework,Version=v4.7.2.AssemblyAttributes.cs

Go to the documentation of this file.

```
// <autogenerated />
using System;
using System.Reflection;
[assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.7.2",
FrameworkDisplayName = ".NET Framework 4.7.2")]
```

6.21 Product/Product.cs File Reference

Classes

- class trabalhoPOO_27967.Product

  Purpose: Definition of product and methods to deal with product operations. Created by: Jose Alves a27967 Created on: 11/2/2024 4:40:12 PM.

Namespaces

- namespace trabalhoPOO_27967

6.22 Product.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/2/2024 4:40:12 PM</date>
*   <description></description>
**/
using System;
using System.Collections;
using System.Collections.Generic;
using System.Diagnostics.Contracts;
using System.Text;
using System.Xml.Linq;

namespace trabalhoPOO_27967
{
    public class Product
    {
        #region Attributes
        string _reference;
        decimal _price;
        Make _make;
        Category _category;
        Warranty _warranty; //in Years
        int _stock;
        #endregion

        #region Methods

        #region Constructors

        public Product()
        {
            _reference = string.Empty;
            _price = -1;
            _make = new Make();
            _category = new Category();
            _stock = 0;
        }

        public Product(string reff, decimal pri, Warranty war, Make ma, Category cat)
        {
            _reference = reff;
            _price = pri;
            _make = ma;
            _category = cat;
            _warranty = war;

        }

        #endregion

        #region Properties

        public string Reference
        {
            get { return _reference; }
            set { _reference = value; }

        }

        public decimal Price
        {
            get { return _price; }
            set { _price = value; }
        }

        public Make Make
        {
            get { return _make; }
            set { _make = value; }

        }

        public Category Category
        {
            get { return _category; }
            set { _category = value; }
        }

        public int Stock
        {
            get { return _stock; }
            set { _stock = value; }
        }

        public Warranty Warranty
        {
            get { return _warranty; }
            set { _warranty = value; }
        }
        #endregion



        #region Overrides
        public override bool Equals(object obj)
        {
            if (obj == null) return false;
            Product product = obj as Product;
            if (product.Reference == _reference) return true;
            return false;
        }

        public static bool operator ==(Product p1, Product p2)
        {
            return (p1.Equals(p2));
        }

        public static bool operator !=(Product p1, Product p2)
        {
            return !(p1.Equals(p2));
        }

        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            sb.AppendLine($"Product ID: {_reference}");
            sb.AppendLine($"Price: {_price}€");
            sb.AppendLine(_make.ToString());
            sb.AppendLine(_warranty.ToString());

            return sb.ToString();
        }

        #endregion

        #region OtherMethods
        #endregion

        #region Destructor
        ~Product()
        {
        }
        #endregion

        #endregion
    }
}
```

6.23 Product/Products.cs File Reference

Classes

- class trabalhoPOO_27967.Products

  Purpose: Class to manage a group of more than one product. Created by: Jose Alves a27967 Created on: 11/9/2024 6:34:19 PM.

Namespaces

- namespace trabalhoPOO_27967

6.24 Products.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/9/2024 6:34:19 PM</date>
*   <description></description>
**/
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using static System.Net.Mime.MediaTypeNames;

namespace trabalhoPOO_27967
{
    public class Products
    {
        #region Attributes
        List<Product> _prods; //como faço uma propriedade para isto? Não consigo dar return do array, certo?
        #endregion

        #region Methods

        #region Constructors

        public Products()
        {
            _prods = new List<Product>();
        }

        public Products(List<Product> products)
        {
            _prods = products;
        }



        #endregion

        #region Properties
        public List<Product> Prods
        {
            get { return _prods; }
            set { _prods = value; }
        }
        #endregion



        #region Overrides

        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            foreach (var product in _prods)
            {
                sb.AppendLine(product.ToString());
            }
            return sb.ToString();
        }
        #endregion

        #region OtherMethods

        public decimal ValueInPosition(int p)
        {
            return this.Prods[p].Price;
        }

        public bool InsertProduct(Product p)
        {
            if (p == null) return false;
            _prods.Add(p);
            return true;
        }

        public bool ExistProduct(string reff)
        {
            foreach (Product p in _prods)
            {
                if (p.Reference == reff) return true;
            }
            return false;
        }

        public bool RemoveProduct(Product p)
        {
            if (p == null) return false;
            if (ExistProduct(p.Reference))
            {
                _prods.Remove(p);
                return true;
            }
            return false;
        }

        public Product SearchProduct(string reff)
        {
            foreach (Product p in _prods)
            {
                if(p.Reference==reff) return p;
            }
            return null;
        }

        public decimal TotalPrice()
        {
            return _prods.Sum(p => p.Price);

        }

        public DateTime WarratyExpirationDateForProduct(DateTime date, string reff)
        {
            Product p = SearchProduct(reff);
            {
                return (date.AddYears(p.Warranty.DurationInYears));
            }
        }
        #endregion

        #region Destructor
        ~Products()
        {
        }
        #endregion

        #endregion
    }
}
```

6.25 Properties/AssemblyInfo.cs File Reference

6.26 AssemblyInfo.cs

Go to the documentation of this file.

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("trabalhoPOO_27967")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("trabalhoPOO_27967")]
[assembly: AssemblyCopyright("Copyright ©  2024")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("c7a4f6de-be70-4f08-92bd-b54d2a0111f1")]

// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

6.27 Sale/Sale.cs File Reference

Classes

- class trabalhoPOO_27967.Sale

  Purpose: Definition of Sale and methods to deal with Sale operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:53 AM.

Namespaces

- namespace trabalhoPOO_27967

6.28 Sale.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/6/2024 11:21:53 AM</date>
*   <description></description>
**/
using System;
using System.CodeDom;
using System.Linq;
using System.Text;

namespace trabalhoPOO_27967
{

    public class Sale
    {
        #region Attributes
        int _id;
        Client _client;
        Products _products;
        decimal _totPrice;
        DateTime _saleDate;
        static int _numSales;
        Campaign _campaigns;
        #endregion


        #region Methods

        #region Constructors

        public Sale()
        {
            _products= new Products();
            _client= new Client();
            _campaigns= new Campaign();
        }

        public Sale(Client client, Products products, Campaign camp)
        {
            _id = ++_numSales;
            _client = client;
            _products = products;
            _totPrice = TotalPrice();
            _saleDate = DateTime.Now;
        }



        #endregion

        #region Properties
        public int Id
        {
            get { return _id; }
            set { _id = value; }
        }

        public Client Client
        {
            get { return (_client); }
            set { _client = value; }
        }

        public Products Products
        {
            get { return _products; }
            set { _products = value; }
        }

        public decimal TotPrice
        {
            get { return _totPrice; }
            set { _totPrice = value; }
        }

        public DateTime SaleDate
        {
            get { return _saleDate; }
            set { _saleDate = value; }
        }

        public Campaign Campaigns
        {
            get { return _campaigns; }
            set { _campaigns = value; }
        }
        #endregion



        #region Overrides
        public override bool Equals(object obj)
        {
            Sale sale = obj as Sale;
            return (sale.Id==this.Id);
        }

        public static bool operator ==(Sale s1, Sale s2)
        {
            return(s1.Equals(s2));
        }

        public static bool operator !=(Sale s1, Sale s2)
        {
            return !(s1.Equals(s2));
        }

        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            sb.AppendLine(_products.ToString());
            sb.AppendLine("Total " + this.TotalPrice().ToString() + "\u20AC");
            return sb.ToString();

        }
        #endregion

        #region OtherMethods

        public decimal TotalPrice()
        {
            decimal total = 0;

            total=_products.TotalPrice();

            if (Campaign.VerifyApplicability(this.Campaigns))
            {
                total *= (1-this.Campaigns.Discount);
            }

            return total;
        }

        public bool InsertProductOnSale(Product p)
        {
            return _products.InsertProduct(p);
        }

        public bool RemoveProductFromSale(Product p)
        {
            return _products.RemoveProduct(p);
        }

        public bool ExistProductOnSale(Product p)
        {
            return _products.ExistProduct(p.Reference);
        }

        public DateTime WarrantyExpirationDate(string reff)
        {
            return (_products.WarratyExpirationDateForProduct(this.SaleDate, reff));
        }
        #endregion

        #region Destructor
        ~Sale()
        {
        }
        #endregion

        #endregion
    }
}
```

6.29 Sale/Sales.cs File Reference

Classes

- class trabalhoPOO_27967.Sales

    Purpose: Class with the agregation of sales of a store. Created by: Jose Alves a27967 Created on: 11/10/2024 7:42:03 PM.

Namespaces

- namespace trabalhoPOO_27967

6.30 Sales.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/10/2024 7:42:03 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;

namespace trabalhoPOO_27967
{
    public class Sales
    {
        #region Attributes
        static List<Sale> _salesStored;
        #endregion

        #region Methods

        #region Constructors

        public Sales()
        {
            _salesStored = new List<Sale>();
        }

        public Sales(List<Sale> sales)
        {
            _salesStored = sales;
        }


        #endregion

        #region Properties
        public List<Sale> SalesStored
        {
            get { return _salesStored; }
            set { _salesStored = value; }
        }

        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        public Sale GetSale(int idSale)
        {
            foreach (Sale s in _salesStored)
            {
                if (s.Id == idSale) return s;
            }
            return null;
        }

        public bool AddSale(Sale sale)
        {
            if (_salesStored == null) return false;
            _salesStored.Add(sale);
            return true;
        }
        #endregion

        #region Destructor
        ~Sales()
        {
        }
        #endregion

        #endregion
    }
}
```

6.31 Store/Store.cs File Reference

Classes

- class trabalhoPOO_27967.Store.Store

  Purpose: This class has the definition and properties to manage a store. Created by: Jose Alves a27967 Created on: 11/14/2024 5:01:23 PM.

Namespaces

- namespace trabalhoPOO_27967
- namespace trabalhoPOO_27967.Store

6.32 Store.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.Store.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/14/2024 5:01:23 PM</date>
*   <description></description>
**/
using System;

namespace trabalhoPOO_27967.Store
{
    public class Store
    {
        #region Attributes
        static Clients _clientList;
        static Products _prodList;
        static Sales _saleList;
        static Makes _makeList;
        static Categories _catList;
        static Warranties _warrantList;
        #endregion

        #region Methods

        #region Constructors

        public Store()
        {
            _clientList = new Clients();
            _prodList = new Products();
            _saleList = new Sales();
            _makeList = new Makes();
            _catList = new Categories();
            _warrantList = new Warranties();
        }

        public Store(Clients cl, Products p, Sales s, Makes m, Categories c, Warranties w)
        {
            _clientList=cl;
            _prodList=p;
            _saleList=s;
            _makeList=m;
            _catList=c;
            _warrantList=w;
        }

        #endregion

        #region Properties
        public Clients ClientLIst
        {
            get { return _clientList; }
            set { _clientList = value; }
        }

        public Products ProdList
        {
            get { return _prodList; }
            set { _prodList = value; }
        }

        public Sales SaleList
        {
            get { return _saleList; }
            set { _saleList = value; }
        }

        public Makes MakeList
        {
            get { return _makeList; }
            set { _makeList = value; }
        }

        public Categories CatList
        {
            get { return _catList; }
            set { _catList = value; }
        }

        public Warranties WarrantList
        {
            get { return _warrantList; }
            set { _warrantList = value; }
        }


        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        #endregion

        #region Destructor
        ~Store()
        {
        }
        #endregion

        #endregion
    }
}
```

6.33 Warranty/Warranties.cs File Reference

Classes

- class trabalhoPOO_27967.Warranties

  Purpose:This file has the definition and methods to work with the plurality of Warranty. Created by: Jose Alves a27967 Created on: 11/14/2024 4:20:11 PM.

Namespaces

- namespace trabalhoPOO_27967

6.34 Warranties.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.Category.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/14/2024 4:20:11 PM</date>
*   <description></description>
**/
using System;
using System.Collections.Generic;

namespace trabalhoPOO_27967
{
    public class Warranties
    {
        #region Attributes
        List<Warranty> _warrants;
        #endregion

        #region Methods

        #region Constructors

        public Warranties()
        {
            _warrants = new List<Warranty>();
        }

        public Warranties(List<Warranty> warrants)
        {
            _warrants = warrants;
        }



        #endregion

        #region Properties
        public List<Warranty> Warrants
        {
            get { return _warrants; }
            set { _warrants = value; }
        }
        #endregion



        #region Overrides
        #endregion

        #region OtherMethods
        // <summary>
        public bool AddWarranty(Warranty w)
        {
            if (w == null) return false;
            _warrants.Add(w);
            return true;
        }

        public bool RemoveWarranty(Warranty w)
        {
            foreach (Warranty war in _warrants)
            {
                if (war==w)
                {
                    _warrants.Remove(war);
                    return true;
                }
            }
            return false;
        }

        public bool ExistWarranty(string id)
        {
            foreach (Warranty w in _warrants)
            {
                if (w.ProdID == id)
                {
                    return true;
                }
            }
            return false;
        }
        #endregion

        #region Destructor
        ~Warranties()
        {
        }
        #endregion

        #endregion
    }
}
```

6.35 Warranty/Warranty.cs File Reference

Classes

- class trabalhoPOO_27967.Warranty

  Purpose: This class contains the definition and methods to manage warranties. Created by: Jose Alves a27967 Created on: 11/13/2024 4:17:18 PM.

Namespaces

- namespace trabalhoPOO_27967

6.36 Warranty.cs

Go to the documentation of this file.

```
/*
*   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
*       Copyright (c) 2024 All Rights Reserved
*   </copyright>
*   <author>Jose Alves a27967</author>
*   <date>11/13/2024 4:17:18 PM</date>
*   <description></description>
**/
using System;
using System.Diagnostics.Contracts;
using System.Text;
using System.Xml.Linq;

namespace trabalhoPOO_27967
{
    public class Warranty
    {
        #region Attributes
        string _prodID;
        int _durationInYears;
        string _conditions;
        #endregion

        #region Methods

        #region Constructors

        public Warranty()
        {
            _prodID = string.Empty;
            _durationInYears = -1;
            _conditions = string.Empty;
        }

        public Warranty(string prodID, int durationInYears, string conditions)
        {
            _prodID = prodID;
            _durationInYears = durationInYears;
            _conditions = conditions;
        }



        #endregion

        #region Properties

        public string ProdID
        {
            get {  return _prodID; }
            set { _prodID = value; }
        }

        public int DurationInYears
        {
            get { return _durationInYears; }
            set { _durationInYears = value; }
        }

        public string Conditions
        {
            get { return _conditions; }
            set { _conditions = value; }
        }
        #endregion



        #region Overrides
        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            sb.AppendLine($"Warranty Data: ");
            sb.AppendLine($"Duration: {_durationInYears} years");
            sb.AppendLine($"Terms: {_conditions}");

            return sb.ToString();
        }

        public override bool Equals(object obj)
        {
            if (obj == null)
            {
                return false;
            }

            Warranty war = obj as Warranty;
            return (this.ProdID == war.ProdID && this.Conditions == war.Conditions);
        }

        public static bool operator ==(Warranty w1, Warranty w2)
        {
            return (w1.Equals(w2));
        }

        public static bool operator !=(Warranty w1, Warranty w2)
        {
            return !(w1.Equals(w2));
        }
        #endregion

        #region OtherMethods
        public DateTime ExpirationDate(Sale s, string reff)
        {
            Product p = s.Products.SearchProduct(reff);
            return (s.SaleDate.AddYears(_durationInYears));
        }
        #endregion

        #region Destructor
        ~Warranty()
        {
        }
        #endregion

        #endregion
    }
}
```

Index