

LESI_TP_POO_27967

v 1.0

Generated by Doxygen 1.9.8

1 Namespace Index	1
1.1 Package List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 trabalhoPOO_27967 Namespace Reference	9
5.2 trabalhoPOO_27967.Interface Namespace Reference	10
5.3 trabalhoPOO_27967.Store Namespace Reference	10
6 Class Documentation	11
6.1 trabalhoPOO_27967.BestSale Class Reference	11
6.1.1 Detailed Description	11
6.2 trabalhoPOO_27967.Campaign Class Reference	11
6.2.1 Detailed Description	12
6.2.2 Constructor & Destructor Documentation	12
6.2.2.1 Campaign() [1/2]	12
6.2.2.2 Campaign() [2/2]	12
6.2.3 Member Function Documentation	13
6.2.3.1 Equals()	13
6.2.3.2 operator!=(())	13
6.2.3.3 operator==(())	13
6.2.3.4 VerifyApplicability()	15
6.2.4 Property Documentation	15
6.2.4.1 CampaignCount	15
6.2.4.2 Discount	15
6.2.4.3 EndDate	16
6.2.4.4 Id	16
6.2.4.5 Name	16
6.2.4.6 StartDate	16
6.3 trabalhoPOO_27967.Campaigns Class Reference	16
6.4 trabalhoPOO_27967.Categories Class Reference	17
6.4.1 Detailed Description	17
6.4.2 Constructor & Destructor Documentation	17
6.4.2.1 Categories() [1/2]	17
6.4.2.2 Categories() [2/2]	17
6.4.3 Member Function Documentation	18

6.4.3.1 Add()	18
6.4.3.2 Exist()	18
6.4.3.3 Remove()	19
6.4.4 Property Documentation	19
6.4.4.1 Cats	19
6.5 trabalhoPOO_27967.Category Class Reference	19
6.5.1 Detailed Description	20
6.5.2 Constructor & Destructor Documentation	20
6.5.2.1 Category() [1/2]	20
6.5.2.2 Category() [2/2]	20
6.5.3 Member Function Documentation	21
6.5.3.1 Equals()	21
6.5.3.2 operator!=(())	21
6.5.3.3 operator==(())	21
6.5.4 Property Documentation	22
6.5.4.1 Id	22
6.5.4.2 Name	22
6.6 trabalhoPOO_27967.Client Class Reference	22
6.6.1 Detailed Description	23
6.6.2 Constructor & Destructor Documentation	23
6.6.2.1 Client() [1/2]	23
6.6.2.2 Client() [2/2]	23
6.6.3 Member Function Documentation	24
6.6.3.1 Equals()	24
6.6.3.2 operator!=(())	24
6.6.3.3 operator==(())	24
6.6.3.4 ToString()	26
6.6.4 Property Documentation	26
6.6.4.1 ClientCount	26
6.6.4.2 ClientID	26
6.6.4.3 Contact	26
6.6.4.4 Name	27
6.7 trabalhoPOO_27967.Clients Class Reference	27
6.7.1 Detailed Description	27
6.7.2 Constructor & Destructor Documentation	28
6.7.2.1 Clients()	28
6.7.3 Member Function Documentation	28
6.7.3.1 Add()	28
6.7.3.2 Exist()	28
6.7.3.3 GetClient()	29
6.7.3.4 Remove()	29
6.7.4 Property Documentation	29

6.7.4.1 ClientList	29
6.8 trabalhoPOO_27967.Interface.IListManagement Interface Reference	30
6.8.1 Detailed Description	30
6.8.2 Member Function Documentation	30
6.8.2.1 Add()	30
6.8.2.2 Exist()	30
6.8.2.3 Remove()	30
6.9 trabalhoPOO_27967.Make Class Reference	31
6.9.1 Detailed Description	31
6.9.2 Constructor & Destructor Documentation	31
6.9.2.1 Make() [1/2]	31
6.9.2.2 Make() [2/2]	31
6.9.3 Member Function Documentation	32
6.9.3.1 Equals()	32
6.9.3.2 operator!=(())	32
6.9.3.3 operator==(())	33
6.9.3.4 ToString()	33
6.9.4 Property Documentation	33
6.9.4.1 ID	33
6.9.4.2 Name	33
6.10 trabalhoPOO_27967.Makes Class Reference	34
6.10.1 Detailed Description	34
6.10.2 Constructor & Destructor Documentation	34
6.10.2.1 Makes() [1/2]	34
6.10.2.2 Makes() [2/2]	34
6.10.3 Member Function Documentation	35
6.10.3.1 Add()	35
6.10.3.2 Exist()	35
6.10.3.3 Remove()	36
6.10.4 Property Documentation	36
6.10.4.1 MakeList	36
6.11 trabalhoPOO_27967.Product Class Reference	36
6.11.1 Detailed Description	37
6.11.2 Constructor & Destructor Documentation	37
6.11.2.1 Product() [1/2]	37
6.11.2.2 Product() [2/2]	37
6.11.3 Member Function Documentation	38
6.11.3.1 Equals()	38
6.11.3.2 operator!=(())	38
6.11.3.3 operator==(())	39
6.11.3.4 ToString()	39
6.11.4 Property Documentation	39

6.11.4.1 Category	39
6.11.4.2 Make	39
6.11.4.3 Price	40
6.11.4.4 Reference	40
6.11.4.5 Stock	40
6.11.4.6 Warranty	40
6.12 trabalhoPOO_27967.Products Class Reference	40
6.12.1 Detailed Description	41
6.12.2 Constructor & Destructor Documentation	41
6.12.2.1 Products() [1/2]	41
6.12.2.2 Products() [2/2]	41
6.12.3 Member Function Documentation	42
6.12.3.1 Add()	42
6.12.3.2 Exist()	42
6.12.3.3 Remove()	43
6.12.3.4 SearchProduct()	43
6.12.3.5 ToString()	43
6.12.3.6 TotalPrice()	44
6.12.3.7 ValueInPosition()	44
6.12.3.8 WarratyExpirationDateForProduct()	44
6.12.4 Property Documentation	45
6.12.4.1 Prods	45
6.13 trabalhoPOO_27967.Sale Class Reference	45
6.13.1 Detailed Description	46
6.13.2 Constructor & Destructor Documentation	46
6.13.2.1 Sale() [1/2]	46
6.13.2.2 Sale() [2/2]	46
6.13.3 Member Function Documentation	47
6.13.3.1 Equals()	47
6.13.3.2 ExistProductOnSale()	47
6.13.3.3 InsertProductOnSale()	47
6.13.3.4 operator!=(())	48
6.13.3.5 operator==(())	48
6.13.3.6 RemoveProductFromSale()	48
6.13.3.7 ToString()	49
6.13.3.8 TotalPrice()	49
6.13.3.9 WarrantyExpirationDate()	49
6.13.4 Property Documentation	50
6.13.4.1 Campaigns	50
6.13.4.2 Client	50
6.13.4.3 Id	50
6.13.4.4 Products	50

6.13.4.5 SaleDate	50
6.13.4.6 TotPrice	50
6.14 trabalhoPOO_27967.Sales Class Reference	51
6.14.1 Detailed Description	51
6.14.2 Constructor & Destructor Documentation	51
6.14.2.1 Sales() [1/2]	51
6.14.2.2 Sales() [2/2]	51
6.14.3 Member Function Documentation	52
6.14.3.1 Add()	52
6.14.3.2 Exist()	52
6.14.3.3 GetSale()	53
6.14.3.4 Remove()	53
6.14.4 Property Documentation	53
6.14.4.1 SalesStored	53
6.15 trabalhoPOO_27967.Store.Store Class Reference	54
6.15.1 Detailed Description	54
6.15.2 Constructor & Destructor Documentation	54
6.15.2.1 Store() [1/2]	54
6.15.2.2 Store() [2/2]	54
6.15.3 Property Documentation	55
6.15.3.1 CatList	55
6.15.3.2 ClientList	55
6.15.3.3 MakeList	55
6.15.3.4 ProdList	55
6.15.3.5 SaleList	56
6.15.3.6 WarrantList	56
6.16 trabalhoPOO_27967.Warranties Class Reference	56
6.16.1 Detailed Description	57
6.16.2 Constructor & Destructor Documentation	57
6.16.2.1 Warranties() [1/2]	57
6.16.2.2 Warranties() [2/2]	57
6.16.3 Member Function Documentation	57
6.16.3.1 Add()	57
6.16.3.2 Exist()	58
6.16.3.3 Remove()	58
6.16.4 Property Documentation	58
6.16.4.1 Warrants	58
6.17 trabalhoPOO_27967.Warranty Class Reference	59
6.17.1 Detailed Description	59
6.17.2 Constructor & Destructor Documentation	59
6.17.2.1 Warranty() [1/2]	59
6.17.2.2 Warranty() [2/2]	60

6.17.3 Member Function Documentation	60
6.17.3.1 Equals()	60
6.17.3.2 ExpirationDate()	60
6.17.3.3 operator"!=()	61
6.17.3.4 operator==()	61
6.17.3.5 ToString()	62
6.17.4 Property Documentation	62
6.17.4.1 Conditions	62
6.17.4.2 DurationInYears	62
6.17.4.3 ProdID	62
7 File Documentation	63
7.1 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/BestSale.cs File Reference	63
7.2 BestSale.cs	63
7.3 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaign.cs File Reference	64
7.4 Campaign.cs	64
7.5 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaigns.cs File Reference	66
7.6 Campaigns.cs	66
7.7 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Categories.cs File Reference	67
7.8 Categories.cs	68
7.9 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Category.cs File Reference	69
7.10 Category.cs	69
7.11 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Client.cs File Reference	70
7.12 Client.cs	71
7.13 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Clients.cs File Reference	72
7.14 Clients.cs	73
7.15 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Interface/IListManagement.cs File Reference	74
7.16 IListManagement.cs	74
7.17 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Make.cs File Reference	75
7.18 Make.cs	75
7.19 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Makes.cs File Reference	76
7.20 Makes.cs	76
7.21 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs File Reference	78
7.22 .NETFramework,Version=v4.7.2.AssemblyAttributes.cs	78
7.23 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Product.cs File Reference	78
7.24 Product.cs	78
7.25 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Products.cs File Reference	80
7.26 Products.cs	80
7.27 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Properties/AssemblyInfo.cs File Reference	82
7.28 AssemblyInfo.cs	82
7.29 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sale.cs File Reference	83
7.30 Sale.cs	83

7.31 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sales.cs File Reference	85
7.32 Sales.cs	85
7.33 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Store/Store.cs File Reference	87
7.34 Store.cs	87
7.35 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranties.cs File Reference	89
7.36 Warranties.cs	89
7.37 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranty.cs File Reference	90
7.38 Warranty.cs	90
Index	93

Chapter 1

Namespace Index

1.1 Package List

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

trabalhoPOO_27967	9
trabalhoPOO_27967.Interface	10
trabalhoPOO_27967.Store	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

trabalhoPOO_27967.BestSale	11
trabalhoPOO_27967.Campaign	11
trabalhoPOO_27967.Category	19
trabalhoPOO_27967.Client	22
trabalhoPOO_27967.Interface.IListManagement	30
trabalhoPOO_27967.Campaigns	16
trabalhoPOO_27967.Categories	17
trabalhoPOO_27967.Clients	27
trabalhoPOO_27967.Makes	34
trabalhoPOO_27967.Products	40
trabalhoPOO_27967.Sales	51
trabalhoPOO_27967.Warranties	56
trabalhoPOO_27967.Make	31
trabalhoPOO_27967.Product	36
trabalhoPOO_27967.Sale	45
trabalhoPOO_27967.Store.Store	54
trabalhoPOO_27967.Warranty	59

Chapter 3

Class Index

3.1 Class List

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

trabalhoPOO_27967.BestSale	11
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	11
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	16
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	17
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	19
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	22
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	27
trabalhoPOO_27967.Interface.IListManagement	30
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	31
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	34
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	36
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	40
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	45

[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 51

[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 54

[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 56

[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 59

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/BestSale.cs	63
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaign.cs	64
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaigns.cs	66
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Categories.cs	67
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Category.cs	69
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Client.cs	70
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Clients.cs	72
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Interface/IListManagement.cs	74
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Make.cs	75
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Makes.cs	76
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs	78
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Product.cs	78
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Products.cs	80
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Properties/AssemblyInfo.cs	82
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sale.cs	83
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sales.cs	85
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Store/Store.cs	87
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranties.cs	89
D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranty.cs	90

Chapter 5

Namespace Documentation

5.1 trabalhoPOO_27967 Namespace Reference

Namespaces

- namespace [Interface](#)
- 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.

5.2 trabalhoPOO_27967.Interface Namespace Reference

Classes

- interface [IListManagement](#)

5.3 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 6

Class Documentation

6.1 trabalhoPOO_27967.BestSale Class Reference

6.1.1 Detailed Description

Definition at line 10 of file [BestSale.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/[BestSale.cs](#)

6.2 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 certain 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.

6.2.1 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](#).

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Campaign() [1/2]

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

The default Constructor.

Definition at line 38 of file [Campaign.cs](#).

6.2.2.2 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

<i>id</i>	
<i>name</i>	
<i>discount</i>	
<i>startDate</i>	
<i>endDate</i>	

Definition at line 55 of file [Campaign.cs](#).

6.2.3 Member Function Documentation

6.2.3.1 Equals()

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

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

Parameters

<i>obj</i>	
------------	--

Returns

Veriffies if the object given is null.

Casts the object to be Campaign.

Definition at line 134 of file [Campaign.cs](#).

6.2.3.2 operator"!=()"

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

Redefinition of the != operator.

Parameters

<i>cli1</i>	
<i>cli2</i>	

Returns

Definition at line 164 of file [Campaign.cs](#).

6.2.3.3 operator==(())

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

Redefinition of the == operator.

Parameters

<i>cli1</i>	
<i>cli2</i>	

Returns

Definition at line 153 of file [Campaign.cs](#).

6.2.3.4 VerifyApplicability()

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

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

Parameters

<i>camp</i>	
-------------	--

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](#).

6.2.4 Property Documentation**6.2.4.1 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](#).

6.2.4.2 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](#).

6.2.4.3 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](#).

6.2.4.4 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](#).

6.2.4.5 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](#).

6.2.4.6 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/[Campaign.cs](#)

6.3 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.

Inheritance diagram for trabalhoPOO_27967.Campaigns:

6.4 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.

Inheritance diagram for trabalhoPOO_27967.Categories:

Collaboration diagram for trabalhoPOO_27967.Categories:

Public Member Functions

- [Categories](#) ()
The default Constructor.
- [Categories](#) (List< [Category](#) > cats)
The constructor to use when a list of categories is given.
- bool [Add](#) (object obj)
Method used to add a category to a list of categories.
- bool [Remove](#) (object obj)
Method used to remove a category from a list of categories.
- bool [Exist](#) (object obj)
Method used to verify if a category exists on a list of makes, given its ID or name.

Properties

- List< [Category](#) > [Cats](#) [get, set]
The property used to get and set the list of categories.

6.4.1 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 23 of file [Categories.cs](#).

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Categories() [1/2]

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

The default Constructor.

Definition at line 36 of file [Categories.cs](#).

6.4.2.2 Categories() [2/2]

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

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

Parameters

<i>cats</i>	
-------------	--

Definition at line 45 of file [Categories.cs](#).

6.4.3 Member Function Documentation

6.4.3.1 Add()

```
bool trabalhoPOO_27967.Categories.Add (  
    object obj )
```

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

Parameters

<i>c</i>	
----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 75 of file [Categories.cs](#).

6.4.3.2 Exist()

```
bool trabalhoPOO_27967.Categories.Exist (  
    object obj )
```

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

Parameters

<i>id</i>	
-----------	--

Returns

If the ID is given

The Name is given

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 109 of file [Categories.cs](#).

6.4.3.3 Remove()

```
bool trabalhoPOO_27967.Categories.Remove (
    object obj )
```

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

Parameters

c	
---	--

Returns

Implements [trabalhoPOO_27967.Interface.ICollectionManagement](#).

Definition at line 91 of file [Categories.cs](#).

6.4.4 Property Documentation

6.4.4.1 Cats

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

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

Definition at line 57 of file [Categories.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/[Categories.cs](#)

6.5 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 category 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.

6.5.1 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](#).

6.5.2 Constructor & Destructor Documentation

6.5.2.1 `Category()` [1/2]

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

The default Constructor.

Definition at line 35 of file [Category.cs](#).

6.5.2.2 `Category()` [2/2]

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

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

Parameters

<code>name</code>	
-------------------	--

Definition at line 45 of file [Category.cs](#).

6.5.3 Member Function Documentation

6.5.3.1 Equals()

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

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

Parameters

<i>obj</i>	
------------	--

Returns

Veriffies if the object given is null.

Casts the object to be Category.

Definition at line 81 of file [Category.cs](#).

6.5.3.2 operator"!=()"

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

Redefinition of the Not Equal Operator.

Parameters

<i>cat1</i>	
<i>cat2</i>	

Returns

Definition at line 111 of file [Category.cs](#).

6.5.3.3 operator==(())

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

Redefinition of the Equal operator.

Parameters

<i>cat1</i>	
<i>cat2</i>	

Returns

Definition at line 100 of file [Category.cs](#).

6.5.4 Property Documentation

6.5.4.1 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](#).

6.5.4.2 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/[Category.cs](#)

6.6 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.

6.6.1 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](#).

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Client() [1/2]

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

The default Constructor.

Definition at line 39 of file [Client.cs](#).

6.6.2.2 Client() [2/2]

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

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

Parameters

<i>n</i>	
<i>c</i>	

Definition at line 51 of file [Client.cs](#).

6.6.3 Member Function Documentation

6.6.3.1 Equals()

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

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

Parameters

<i>obj</i>	
------------	--

Returns

Veriffies if the object given is null.

Casts the object to be Client.

Definition at line 131 of file [Client.cs](#).

6.6.3.2 operator"!=()"

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

Redefinition of the != operator.

Parameters

<i>cli1</i>	
<i>cli2</i>	

Returns

Definition at line 161 of file [Client.cs](#).

6.6.3.3 operator==(())

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

Redefinition of the == operator.

Parameters

<i>cli1</i>	
<i>cli2</i>	

Returns

Definition at line 150 of file [Client.cs](#).

6.6.3.4 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](#).

6.6.4 Property Documentation**6.6.4.1 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](#).

6.6.4.2 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](#).

6.6.4.3 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](#).

6.6.4.4 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/[Client.cs](#)

6.7 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.

Inheritance diagram for trabalhoPOO_27967.Clients:

Collaboration diagram for trabalhoPOO_27967.Clients:

Public Member Functions

- [Clients](#) ()
The default Constructor.
- bool [Add](#) (object obj)
Method to add a client to the store's client list.
- bool [Remove](#) (object obj)
Method to remove a client from the store's client list.
- bool [Exist](#) (object obj)
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.

6.7.1 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 22 of file [Clients.cs](#).

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Clients()

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

The default Constructor.

Definition at line 35 of file [Clients.cs](#).

6.7.3 Member Function Documentation

6.7.3.1 Add()

```
bool trabalhoPOO_27967.Clients.Add (
    object obj )
```

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

Parameters

<i>cli</i>	
------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 66 of file [Clients.cs](#).

6.7.3.2 Exist()

```
bool trabalhoPOO_27967.Clients.Exist (
    object obj )
```

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

Parameters

<i>id</i>	
-----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 99 of file [Clients.cs](#).

6.7.3.3 GetClient()

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

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

Parameters

<i>id</i>	
-----------	--

Returns

Definition at line 119 of file [Clients.cs](#).

6.7.3.4 Remove()

```
bool trabalhoPOO_27967.Clients.Remove (
    object obj )
```

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

Parameters

<i>cli</i>	
------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 82 of file [Clients.cs](#).

6.7.4 Property Documentation

6.7.4.1 ClientList

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

Property used to get and set the client list.

Definition at line 48 of file [Clients.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/[Clients.cs](#)

6.8 trabalhoPOO_27967.Interface.IListManagement Interface Reference

Inheritance diagram for trabalhoPOO_27967.Interface.IListManagement:

Public Member Functions

- bool [Add](#) (object obj)
- bool [Remove](#) (object obj)
- bool [Exist](#) (object obj)

6.8.1 Detailed Description

Definition at line 17 of file [IListManagement.cs](#).

6.8.2 Member Function Documentation

6.8.2.1 Add()

```
bool trabalhoPOO_27967.Interface.IListManagement.Add (  
    object obj )
```

Implemented in [trabalhoPOO_27967.Campaigns](#), [trabalhoPOO_27967.Categories](#), [trabalhoPOO_27967.Clients](#), [trabalhoPOO_27967.Makes](#), [trabalhoPOO_27967.Products](#), [trabalhoPOO_27967.Sales](#), and [trabalhoPOO_27967.Warranties](#).

6.8.2.2 Exist()

```
bool trabalhoPOO_27967.Interface.IListManagement.Exist (  
    object obj )
```

Implemented in [trabalhoPOO_27967.Campaigns](#), [trabalhoPOO_27967.Categories](#), [trabalhoPOO_27967.Clients](#), [trabalhoPOO_27967.Makes](#), [trabalhoPOO_27967.Products](#), [trabalhoPOO_27967.Sales](#), and [trabalhoPOO_27967.Warranties](#).

6.8.2.3 Remove()

```
bool trabalhoPOO_27967.Interface.IListManagement.Remove (  
    object obj )
```

Implemented in [trabalhoPOO_27967.Campaigns](#), [trabalhoPOO_27967.Categories](#), [trabalhoPOO_27967.Clients](#), [trabalhoPOO_27967.Makes](#), [trabalhoPOO_27967.Products](#), [trabalhoPOO_27967.Sales](#), and [trabalhoPOO_27967.Warranties](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Interface/[IListManagement.cs](#)

6.9 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.

6.9.1 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](#).

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Make() [1/2]

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

The default Constructor.

Definition at line 38 of file [Make.cs](#).

6.9.2.2 Make() [2/2]

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

Constructor when the name of the make is given.

Parameters

<i>id</i>	
<i>name</i>	

Definition at line 49 of file [Make.cs](#).

6.9.3 Member Function Documentation

6.9.3.1 Equals()

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

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

Parameters

<i>obj</i>	
------------	--

Returns

Veriffies if the object given is null.

Casts the object to be Make.

Definition at line 98 of file [Make.cs](#).

6.9.3.2 operator"!=()

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

The redefinition of the NOt Equal operator.

Parameters

<i>m1</i>	
<i>m2</i>	

Returns

Definition at line 128 of file [Make.cs](#).

6.9.3.3 operator==()

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

The redefinition of the Equal operator.

Parameters

<i>m1</i>	
<i>m2</i>	

Returns

Definition at line 117 of file [Make.cs](#).

6.9.3.4 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](#).

6.9.4 Property Documentation

6.9.4.1 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](#).

6.9.4.2 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/[Make.cs](#)

6.10 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.

Inheritance diagram for trabalhoPOO_27967.Makes:

Collaboration diagram for trabalhoPOO_27967.Makes:

Public Member Functions

- [Makes](#) ()
The default Constructor.
- [Makes](#) (List< [Make](#) > m)
The constructor to use when a list of Make is given.
- bool [Add](#) (object obj)
Method used to add a make to a list of makes.
- bool [Remove](#) (object obj)
Method used to remove a make from a list of makes.
- bool [Exist](#) (object obj)
Method used to verify if a make exists on a list of makes, given its ID or name.

Properties

- List< [Make](#) > [MakeList](#) [get, set]
The property to get and set a list of Make.

6.10.1 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 23 of file [Makes.cs](#).

6.10.2 Constructor & Destructor Documentation

6.10.2.1 Makes() [1/2]

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

The default Constructor.

Definition at line 36 of file [Makes.cs](#).

6.10.2.2 Makes() [2/2]

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

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

Parameters

<i>m</i>	
----------	--

Definition at line 45 of file [Makes.cs](#).

6.10.3 Member Function Documentation

6.10.3.1 Add()

```
bool trabalhoPOO_27967.Makes.Add (  
    object obj )
```

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

Parameters

<i>m</i>	
----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 73 of file [Makes.cs](#).

6.10.3.2 Exist()

```
bool trabalhoPOO_27967.Makes.Exist (  
    object obj )
```

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

Parameters

<i>id</i>	
-----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 105 of file [Makes.cs](#).

6.10.3.3 Remove()

```
bool trabalhoPOO_27967.Makes.Remove (
    object obj )
```

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

Parameters

<i>m</i>	
----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 88 of file [Makes.cs](#).

6.10.4 Property Documentation

6.10.4.1 MakeList

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

The property to get and set a list of Make.

Definition at line 55 of file [Makes.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/[Makes.cs](#)

6.11 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]

6.11.1 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`.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 `Product()` [1/2]

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

The default Constructor.

Definition at line 43 of file `Product.cs`.

6.11.2.2 `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

<i>reff</i>	
<i>pri</i>	

Definition at line 57 of file [Product.cs](#).

6.11.3 Member Function Documentation

6.11.3.1 Equals()

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

Redefinition of the method to compare two products.

Parameters

<i>obj</i>	
------------	--

Returns

Definition at line 133 of file [Product.cs](#).

6.11.3.2 operator"!=()

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

Redefinition of the different operator.

Parameters

<i>p1</i>	
<i>p2</i>	

Returns

Definition at line 158 of file [Product.cs](#).

6.11.3.3 operator==()

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

Redefinition of the equal operator.

Parameters

<i>p1</i>	
<i>p2</i>	

Returns

Definition at line 147 of file [Product.cs](#).

6.11.3.4 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](#).

6.11.4 Property Documentation

6.11.4.1 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](#).

6.11.4.2 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](#).

6.11.4.3 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](#).

6.11.4.4 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](#).

6.11.4.5 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](#).

6.11.4.6 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/[Product.cs](#)

6.12 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.

Inheritance diagram for trabalhoPOO_27967.Products:

Collaboration diagram for trabalhoPOO_27967.Products:

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 [Add](#) (object obj)
This method inserts a product in a list of products.
- bool [Exist](#) (object obj)
Method used to verify if a product is on a products' list, given its Reference.
- bool [Remove](#) (object obj)
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 reference.
- decimal [TotalPrice](#) ()
Method used to calculate the total price of products in a list of products.
- DateTime [WarrantyExpirationDateForProduct](#) (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.

6.12.1 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 26 of file [Products.cs](#).

6.12.2 Constructor & Destructor Documentation

6.12.2.1 Products() [1/2]

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

The default Constructor.

Definition at line 39 of file [Products.cs](#).

6.12.2.2 Products() [2/2]

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

The constructor to use when list of Product is given.

Parameters

<i>products</i>	
-----------------	--

Definition at line 48 of file [Products.cs](#).

6.12.3 Member Function Documentation

6.12.3.1 Add()

```
bool trabalhoPOO_27967.Products.Add (
    object obj )
```

This method inserts a product in a list of products.

Parameters

<i>p</i>	
----------	--

Returns

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

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 104 of file [Products.cs](#).

6.12.3.2 Exist()

```
bool trabalhoPOO_27967.Products.Exist (
    object obj )
```

Method used to verify if a product is on a products' list, given its Reference.

Parameters

<i>reff</i>	
-------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 124 of file [Products.cs](#).

6.12.3.3 Remove()

```
bool trabalhoPOO_27967.Products.Remove (
    object obj )
```

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

Parameters

<i>p</i>	
----------	--

Returns

Product removed successfully

Product was not removed.

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 142 of file [Products.cs](#).

6.12.3.4 SearchProduct()

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

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

Parameters

<i>reff</i>	
-------------	--

Returns

Returns the product if found

Definition at line 159 of file [Products.cs](#).

6.12.3.5 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 76 of file [Products.cs](#).

6.12.3.6 TotalPrice()

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

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

Returns

Definition at line 172 of file [Products.cs](#).

6.12.3.7 ValueInPosition()

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

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

Parameters

<i>p</i>	Position in array.
----------	--------------------

Returns

Definition at line 94 of file [Products.cs](#).

6.12.3.8 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

<i>date</i>	
<i>reff</i>	

Returns

Definition at line 184 of file [Products.cs](#).

6.12.4 Property Documentation

6.12.4.1 Prods

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

Property used to get and set the list of products.

Definition at line 61 of file [Products.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/[Products.cs](#)

6.13 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)
Redefinition 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.

6.13.1 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](#).

6.13.2 Constructor & Destructor Documentation

6.13.2.1 Sale() [1/2]

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

The default Constructor.

Definition at line 45 of file [Sale.cs](#).

6.13.2.2 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

<i>client</i>	
<i>products</i>	
<i>camp</i>	

Definition at line 58 of file [Sale.cs](#).

6.13.3 Member Function Documentation

6.13.3.1 Equals()

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

Redefinition of the Equals method to compare two sales.

Parameters

<i>obj</i>	
------------	--

Returns

Definition at line 135 of file [Sale.cs](#).

6.13.3.2 ExistProductOnSale()

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

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

Parameters

<i>p</i>	
----------	--

Returns

Definition at line 218 of file [Sale.cs](#).

6.13.3.3 InsertProductOnSale()

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

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

Parameters

<i>p</i>	
----------	--

Returns

Definition at line 198 of file [Sale.cs](#).

6.13.3.4 operator"!=()

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

Redefinition of the different operator.

Parameters

<i>s1</i>	
<i>s2</i>	

Returns

Definition at line 158 of file [Sale.cs](#).

6.13.3.5 operator==()

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

Redefiniiton of the equal operator.

Parameters

<i>s1</i>	
<i>s2</i>	

Returns

Definition at line 147 of file [Sale.cs](#).

6.13.3.6 RemoveProductFromSale()

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

Method used to remove a product from a sale.

Parameters

<i>p</i>	
----------	--

Returns

Definition at line 208 of file [Sale.cs](#).

6.13.3.7 ToString()

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

\u20AC é o unicode para o símbolo de euro.

Definition at line 163 of file [Sale.cs](#).

6.13.3.8 TotalPrice()

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

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

Returns

The total price to pay.

Definition at line 179 of file [Sale.cs](#).

6.13.3.9 WarrantyExpirationDate()

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

Method to calculate when a warranty is due to expire.

Parameters

<i>s</i>	
<i>reff</i>	

Returns

Definition at line 229 of file [Sale.cs](#).

6.13.4 Property Documentation

6.13.4.1 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](#).

6.13.4.2 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](#).

6.13.4.3 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](#).

6.13.4.4 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](#).

6.13.4.5 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](#).

6.13.4.6 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/[Sale.cs](#)

6.14 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.

Inheritance diagram for trabalhoPOO_27967.Sales:

Collaboration diagram for trabalhoPOO_27967.Sales:

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 [Add](#) (object obj)
Method used to add a sale to a sales' list.
- bool [Remove](#) (object obj)
Method used to remove a sale from a list of sales.
- bool [Exist](#) (object obj)
Method used to check if a sale exists in a list of sales.

Properties

- List< [Sale](#) > [SalesStored](#) [get, set]
Property used to get and set a list of sales.

6.14.1 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 22 of file [Sales.cs](#).

6.14.2 Constructor & Destructor Documentation

6.14.2.1 Sales() [1/2]

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

The default Constructor.

Definition at line 35 of file [Sales.cs](#).

6.14.2.2 Sales() [2/2]

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

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

Parameters

<i>sales</i>	
--------------	--

Definition at line 44 of file [Sales.cs](#).

6.14.3 Member Function Documentation

6.14.3.1 Add()

```
bool trabalhoPOO_27967.Sales.Add (  
    object obj )
```

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

Parameters

<i>sale</i>	
-------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 89 of file [Sales.cs](#).

6.14.3.2 Exist()

```
bool trabalhoPOO_27967.Sales.Exist (  
    object obj )
```

Method used to check if a sale exists in a list of sales.

Parameters

<i>obj</i>	
------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 125 of file [Sales.cs](#).

6.14.3.3 GetSale()

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

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

Parameters

<i>idSale</i>	
---------------	--

Returns

Definition at line 75 of file [Sales.cs](#).

6.14.3.4 Remove()

```
bool trabalhoPOO_27967.Sales.Remove (
    object obj )
```

Method used to remove a sale from a list of sales.

Parameters

<i>obj</i>	
------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 105 of file [Sales.cs](#).

6.14.4 Property Documentation

6.14.4.1 SalesStored

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

Property used to get and set a list of sales.

Definition at line 56 of file [Sales.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/[Sales.cs](#)

6.15 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.

6.15.1 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](#).

6.15.2 Constructor & Destructor Documentation

6.15.2.1 Store() [1/2]

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

The default Constructor.

Definition at line 38 of file [Store.cs](#).

6.15.2.2 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

<i>cl</i>	
<i>p</i>	
<i>s</i>	
<i>m</i>	
<i>c</i>	
<i>w</i>	

Definition at line 57 of file [Store.cs](#).

6.15.3 Property Documentation

6.15.3.1 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](#).

6.15.3.2 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](#).

6.15.3.3 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](#).

6.15.3.4 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](#).

6.15.3.5 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](#).

6.15.3.6 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Store/[Store.cs](#)

6.16 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.

Inheritance diagram for trabalhoPOO_27967.Warranties:

Collaboration diagram for trabalhoPOO_27967.Warranties:

Public Member Functions

- [Warranties](#) ()
The default Constructor.
- [Warranties](#) (List< [Warranty](#) > warrants)
The constructor to use when a list of Warranties is given.
- bool [Add](#) (object obj)
- bool [Remove](#) (object obj)
Method used to remove a warranty from a list of warranties.
- bool [Exist](#) (object obj)
Method used to confirm if a warranty exists on a list of warranties, given the product ID.

Properties

- List< [Warranty](#) > [Warrants](#) [get, set]
Property used to get and set the list of warranties.

6.16.1 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 22 of file [Warranties.cs](#).

6.16.2 Constructor & Destructor Documentation

6.16.2.1 Warranties() [1/2]

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

The default Constructor.

Definition at line 35 of file [Warranties.cs](#).

6.16.2.2 Warranties() [2/2]

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

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

Parameters

<i>warrants</i>	
-----------------	--

Definition at line 44 of file [Warranties.cs](#).

6.16.3 Member Function Documentation

6.16.3.1 Add()

```
bool trabalhoPOO_27967.Warranties.Add (
    object obj )
```

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

Parameters

<i>c</i>	
----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 75 of file [Warranties.cs](#).

6.16.3.2 Exist()

```
bool trabalhoPOO_27967.Warranties.Exist (
    object obj )
```

Method used to confirm if a warranty exists on a list of warranties, given the product ID.

Parameters

<i>id</i>	
-----------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 111 of file [Warranties.cs](#).

6.16.3.3 Remove()

```
bool trabalhoPOO_27967.Warranties.Remove (
    object obj )
```

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

Parameters

<i>camp</i>	
-------------	--

Returns

Implements [trabalhoPOO_27967.Interface.IListManagement](#).

Definition at line 91 of file [Warranties.cs](#).

6.16.4 Property Documentation

6.16.4.1 Warrants

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

Property used to get and set the list of warranties.

Definition at line 57 of file [Warranties.cs](#).

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

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/[Warranties.cs](#)

6.17 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.

6.17.1 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](#).

6.17.2 Constructor & Destructor Documentation

6.17.2.1 Warranty() [1/2]

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

The default Constructor.

Definition at line 38 of file [Warranty.cs](#).

6.17.2.2 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

<i>prodID</i>	
<i>durationInYears</i>	
<i>conditions</i>	

Definition at line 51 of file [Warranty.cs](#).

6.17.3 Member Function Documentation

6.17.3.1 Equals()

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

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

Parameters

<i>obj</i>	
------------	--

Returns

Verifies if the object given is null.

Casts the object to be Warranty.

Definition at line 114 of file [Warranty.cs](#).

6.17.3.2 ExpirationDate()

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

Method to calculate when a warranty is due to expire.

Parameters

<i>s</i>	
<i>reff</i>	

Returns

Definition at line 157 of file [Warranty.cs](#).

6.17.3.3 operator"!=()

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

Redefinition of the != operator.

Parameters

<i>w1</i>	
<i>w2</i>	

Returns

Definition at line 144 of file [Warranty.cs](#).

6.17.3.4 operator==()

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

Redefinition of the == operator.

Parameters

<i>w1</i>	
<i>w2</i>	

Returns

Definition at line 133 of file [Warranty.cs](#).

6.17.3.5 ToString()

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

Method to show the information of a Warranty.

Returns

Definition at line 99 of file [Warranty.cs](#).

6.17.4 Property Documentation

6.17.4.1 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](#).

6.17.4.2 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](#).

6.17.4.3 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:

- D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/[Warranty.cs](#)

Chapter 7

File Documentation

7.1 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/BestSale.cs File Reference

Classes

- class `trabalhoPOO_27967.BestSale`

Namespaces

- namespace `trabalhoPOO_27967`

7.2 BestSale.cs

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

```
00001 using System;
00002 using System.Collections.Generic;
00003 using System.Linq;
00004 using System.Text;
00005 using System.Threading.Tasks;
00006 using System.Text.RegularExpressions; //Used to verify if a string meets certain criteria.
00007
00008 namespace trabalhoPOO_27967
00009 {
00010     class BestSale
00011     {
00012         static void Main(string[] args)
00013         {
00014             Console.OutputEncoding = System.Text.Encoding.UTF8;
00015
00016             Client clientTest = new Client("Jose Alves","962310421");
00017             Client clientTest2 = new Client("Rui Jordao", "931250420");
00018             Client clientTest1 = new Client();
00019
00020             Make benfica = new Make("Benfica");
00021             Category futebol = new Category("Futebol");
00022             Product product1 = new Product("1A34", 23.9m, new Warranty("1A34",3, "Nao gostar de
batatas."), benfica, futebol );
00023
00024             Make braga = new Make("Braga");
00025             Product product2 = new Product("2V45", 15.9m, new Warranty("2V45", 3, "Nao gostar de
peixe."), braga, futebol);
00026
00027             Sale sale = new Sale();
00028             sale.InsertProductOnSale(product1);
00029             sale.InsertProductOnSale(product2);
00030
```

```

00031         Console.WriteLine(sale.ToString());
00032
00033         Clients clientes = new Clients();
00034         clientes.AddClient(clientTest);
00035         clientes.AddClient(clientTest2);
00036
00037         foreach (Client client in clientes.ClientList)
00038         {
00039             Console.WriteLine(client.ToString());
00040         }
00041     }
00042 }
00043
00044 }
00045
00046 }

```

7.3 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.4 Campaign.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/6/2024 11:21:43 AM</date>
00007  * <description></description>
00008 **/
00009 using System;
00010
00011 namespace trabalhoPOO_27967
00012 {
00020     public class Campaign
00021     {
00022         #region Attributes
00023         int _id;
00024         string _name;
00025         decimal _discount;
00026         DateTime _startDate;
00027         DateTime _endDate;
00028         static int _campaignCount;
00029         #endregion
00030
00031         #region Methods
00032
00033         #region Constructors
00034
00038         public Campaign()
00039         {
00040             _id = ++_campaignCount;
00041             _name = string.Empty;
00042             _discount = 0m;
00043             _startDate = DateTime.MinValue;
00044             _endDate = DateTime.MaxValue;
00045         }

```

```

00046
00055     public Campaign(string name, decimal discount, DateTime startDate, DateTime endDate)
00056     {
00057         _id = ++_campaignCount;
00058         _name = name;
00059         _discount = discount;
00060         _startDate = startDate;
00061         _endDate = endDate;
00062     }
00063
00064
00065
00066
00067
00068     #endregion
00069
00070     #region Properties
00071
00075     public int Id{
00076         get { return _id; }
00077         set { _id = value; }
00078     }
00079
00083     public string Name
00084     {
00085         get { return _name; }
00086         set { _name = value; }
00087     }
00088
00092     public decimal Discount
00093     {
00094         get { return _discount; }
00095         set { _discount = value; }
00096     }
00097
00101     public DateTime StartDate
00102     {
00103         get { return _startDate; }
00104         set { _startDate = value; }
00105     }
00106
00110     public DateTime EndDate
00111     {
00112         get { return _endDate; }
00113         set { _endDate = value; }
00114     }
00115
00119     public int CampaignCount
00120     {
00121         get { return _campaignCount; }
00122         set { _campaignCount = value; }
00123     }
00124     #endregion
00125
00126
00127
00128     #region Overrides
00134     public override bool Equals(object obj)
00135     {
00137         if (obj == null)
00138         {
00139             return false;
00140         }
00141
00143         Campaign camp = obj as Campaign;
00144         return (this.Id == camp.Id && _name == camp.Name);
00145     }
00146
00153     public static bool operator ==(Campaign camp1, Campaign camp2)
00154     {
00155         return (camp1.Equals(camp2));
00156     }
00157
00164     public static bool operator !=(Campaign camp1, Campaign camp2)
00165     {
00166         return !(camp1.Equals(camp2));
00167     }
00168     #endregion
00169
00170     #region OtherMethods
00171
00177     static public bool VerifyApplicability(Campaign camp)
00178     {
00180         if (camp.StartDate <= DateTime.Now && camp.EndDate >= DateTime.Now)
00181         {
00182             return true;
00183         }

```

```

00184
00185         return false;
00186     }
00187     #endregion
00188
00189     #region Destructor
00193     ~Campaign()
00194     {
00195     }
00196     #endregion
00197
00198     #endregion
00199 }
00200 }

```

7.5 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.6 Campaigns.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/14/2024 3:57:04 PM</date>
00007  * <description></description>
00008 **/
00009 using System;
00010 using System.Collections.Generic;
00011 using System.Xml.Linq;
00012 using trabalhoPOO_27967.Interface;
00013
00014 namespace trabalhoPOO_27967
00015 {
00023     public class Campaigns : IListManagement
00024     {
00025         #region Attributes
00026         List<Campaign> _camps;
00027         #endregion
00028
00029         #region Methods
00030
00031         #region Constructors
00032
00036         public Campaigns()
00037         {
00038             _camps = new List<Campaign>();
00039         }
00040
00045         public Campaigns(List<Campaign> p) {
00046             _camps = p;
00047         }
00048
00049         #endregion
00050
00051         #region Properties

```

```

00055     public List<Campaign> Camps
00056     {
00057         get{ return _camps; }
00058         set{ _camps = value; }
00059     }
00060     #endregion
00061
00062
00063
00064     #region Overrides
00065
00066     #endregion
00067
00068     #region OtherMethods
00074     public bool Add(object obj)
00075     {
00076         if(obj==null) return false;
00077         if (obj is Campaign)
00078         {
00079             _camps.Add((Campaign)obj);
00080             return true;
00081         }
00082         return false;
00083     }
00084
00090     public bool Remove(object obj)
00091     {
00092         if(obj==null) return false;
00093         var aux=obj as Campaign;
00094         if (Exist(aux.Id) || Exist(aux.Name))
00095         {
00096
00097             _camps.Remove((Campaign)obj);
00098             return true;
00099         }
00100         return false;
00101     }
00102
00103
00109     public bool Exist(object obj)
00110     {
00111         if (obj == null) return false;
00112         var aux=obj as Campaign;
00113         foreach (Campaign c in _camps)
00114         {
00115             if (c.Id == aux.Id || c.Name==aux.Name)
00116             {
00117                 return true;
00118             }
00119         }
00120         return false;
00121     }
00122     #endregion
00123
00124     #region Destructor
00128     ~Campaigns()
00129     {
00130     }
00131     #endregion
00132
00133     #endregion
00134 }
00135 }

```

7.7 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.8 Categories.cs

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

```

00001  /*
00002  *   <copyright file="trabalhoPOO_27967.Category.cs" company="IPCA">
00003  *       Copyright (c) 2024 All Rights Reserved
00004  *   </copyright>
00005  *   <author>Jose Alves a27967</author>
00006  *   <date>11/14/2024 4:45:58 PM</date>
00007  *   <description></description>
00008  */
00009  using System;
00010  using System.Collections.Generic;
00011  using System.Xml.Linq;
00012  using trabalhoPOO_27967.Interface;
00013
00014  namespace trabalhoPOO_27967
00015  {
00023      public class Categories : IListManagement
00024      {
00025          #region Attributes
00026          List<Category> _cats;
00027          #endregion
00028
00029          #region Methods
00030
00031          #region Constructors
00032
00036          public Categories()
00037          {
00038              _cats = new List<Category>();
00039          }
00040
00045          public Categories(List<Category> cats)
00046          {
00047              _cats = cats;
00048          }
00049
00050
00051          #endregion
00052
00053          #region Properties
00057          public List<Category> Cats
00058          {
00059              get { return _cats; }
00060              set { _cats = value; }
00061          }
00062          #endregion
00063
00064
00065
00066          #region Overrides
00067          #endregion
00068
00069          #region OtherMethods
00075          public bool Add(object obj)
00076          {
00077              if (obj == null) return false;
00078              if (obj is Category)
00079              {
00080                  _cats.Add((Category)obj);
00081                  return true;
00082              }
00083              return false;
00084          }
00085
00091          public bool Remove(object obj)
00092          {
00093              if (obj == null) return false;
00094              var aux = obj as Category;
00095              if (Exist(aux.Id) || Exist(aux.Name))
00096              {
00097                  _cats.Remove((Category)obj);
00098                  return true;
00099              }
00100
00101              return false;
00102          }
00103
00109          public bool Exist(object obj)
00110          {
00111              if (obj == null) return false;
00112              var aux=obj as Category;
00113
00115              if (obj is int)

```

```

00116         {
00117             foreach (Category cate in _cats)
00118             {
00119                 if (cate.Id == aux.Id)
00120                 {
00121                     return true;
00122                 }
00123             }
00124         }
00125
00126         if(obj is string)
00127         {
00128             foreach (Category cate in _cats)
00129             {
00130                 if (cate.Name == aux.Name)
00131                 {
00132                     return true;
00133                 }
00134             }
00135         }
00136         return false;
00137     }
00138 }
00139
00140
00141 #endregion
00142
00143 #region Destructor
00144 ~Categories()
00145 {
00146 }
00147 #endregion
00148
00149 #endregion
00150
00151 #endregion
00152
00153 }
00154 }

```

7.9 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.10 Category.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/6/2024 11:20:47 AM</date>
00007  * <description></description>
00008  */
00009 using System;
00010
00011 namespace trabalhoPOO_27967
00012 {
00020     public class Category
00021     {
00022         #region Attributes
00023         int _id;

```

```

00024     string _name;
00025     static int _catCount=0;
00026     #endregion
00027
00028     #region Methods
00029
00030     #region Constructors
00031
00035     public Category()
00036     {
00037         _id = ++_catCount;
00038         _name = string.Empty;
00039     }
00040
00045     public Category(string name)
00046     {
00047         _id = ++_catCount;
00048         _name = name;
00049     }
00050     #endregion
00051
00052     #region Properties
00053
00057     public int Id
00058     {
00059         get { return _id; }
00060         set { _id = value; }
00061     }
00062
00066     public string Name
00067     {
00068         get { return _name; }
00069         set { _name = value; }
00070     }
00071     #endregion
00072
00073
00074
00075     #region Overrides
00081     public override bool Equals(object obj)
00082     {
00083         if (obj == null)
00084         {
00085             return false;
00086         }
00087
00088         Category cat = obj as Category;
00090         return (this.Id == cat.Id || _name == cat.Name);
00091     }
00092
00093
00100     public static bool operator ==(Category cat1, Category cat2)
00101     {
00102         return (cat1.Equals(cat2));
00103     }
00104
00111     public static bool operator !=(Category cat1, Category cat2)
00112     {
00113         return !(cat1.Equals(cat2));
00114     }
00115     #endregion
00116
00117     #region OtherMethods
00118     #endregion
00119
00120     #region Destructor
00124     ~Category()
00125     {
00126     }
00127     #endregion
00128
00129     #endregion
00130 }
00131 }

```

7.11 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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`

7.12 Client.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>10/29/2024 4:23:56 PM</date>
00007  * <description></description>
00008 **/
00009 using System;
00010 using System.Text;
00011 using System.Text.RegularExpressions;
00012 using static System.Net.Mime.MediaTypeNames;
00013
00014 namespace trabalhoPOO_27967
00015 {
00023     public class Client
00024     {
00025         #region Attributes
00026         int _clientID;
00027         string _name;
00028         string _contact;
00029         static int _clientCount=0;
00030         #endregion
00031
00032         #region Methods
00033
00034         #region Constructors
00035
00039         public Client()
00040         {
00041             _clientID = ++_clientCount ;
00042             _name = "No Name";
00043             _contact = "999999999";
00044         }
00045
00051         public Client(string n, string c)
00052         {
00053             _clientID = ++_clientCount;
00054             _name = n;
00055             _contact = c;
00056             //COMO POSSO FAZER PARA TESTAR SE A STRING PODE SER CONTATCO? DEVO FAZE-LO NO CONSTRUTOR
00057             OU FORA?
00058         }
00059         #endregion
00060
00061         #region Properties
00065         public int ClientID
00066         {
00067             get { return _clientID; }
00068             set { _clientID = value; }
00069         }
00070
00074         public string Name
00075         {
00076             get { return _name; }
00077             set { _name = value; }
00078         }
00079
00083         public string Contact
00084         {
00085             get { return _contact; }
00086             set
00087             {
00088                 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).

```

```

00089         bool isGood = Regex.IsMatch(value, pattern); //Verifies if the value meets the
criteria.
00090
00091         if (isGood) _contact = value;
00092
00093         //COMO POSSO RETORNAR UM ERRO CASO A STRING NAO CORRESPONDA?
00094     }
00095 }
00096
00097 public static int ClientCount
00098 {
00099     get { return _clientCount; }
00100     set { _clientCount = value; }
00101 }
00102 #endregion
00103
00104 #region Overrides
00105
00106 public override string ToString()
00107 {
00108     StringBuilder sb = new StringBuilder();
00109     sb.AppendLine($"Client ID: {_clientID}");
00110     sb.AppendLine($"Name: {_name}");
00111     sb.AppendLine($"Contact: {_contact}");
00112
00113     return sb.ToString();
00114 }
00115
00116 public override bool Equals(object obj)
00117 {
00118     if (obj == null)
00119     {
00120         return false;
00121     }
00122
00123     Client client = obj as Client;
00124     return (this._clientID == client._clientID && _name == client._name);
00125 }
00126
00127 public static bool operator == (Client cli1, Client cli2)
00128 {
00129     return (cli1.Equals(cli2));
00130 }
00131
00132 public static bool operator != (Client cli1, Client cli2)
00133 {
00134     return !(cli1.Equals(cli2));
00135 }
00136 #endregion
00137
00138 #region OtherMethods
00139 #endregion
00140
00141 #region Destructor
00142 ~Client()
00143 {
00144 }
00145 #endregion
00146
00147 #endregion
00148 }
00149 }

```

7.13 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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`

7.14 Clients.cs

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

```

00001  /*
00002  *   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *       Copyright (c) 2024 All Rights Reserved
00004  *   </copyright>
00005  *   <author>Jose Alves a27967</author>
00006  *   <date>11/12/2024 9:25:28 PM</date>
00007  *   <description></description>
00008  */
00009  using System;
00010  using System.Collections.Generic;
00011  using trabalhoPOO_27967.Interface;
00012
00013  namespace trabalhoPOO_27967
00014  {
00022      public class Clients : IListManagement
00023      {
00024          #region Attributes
00025          static List<Client> _clientList;
00026          #endregion
00027
00028          #region Methods
00029
00030          #region Constructors
00031
00035          public Clients()
00036          {
00037              _clientList = new List<Client>();
00038          }
00039
00040
00041          #endregion
00042
00043          #region Properties
00044
00048          public List<Client> ClientList
00049          {
00050              get{ return _clientList; }
00051              set{ _clientList = value; }
00052          }
00053          #endregion
00054
00055
00056          #region Overrides
00057          #endregion
00058
00059          #region OtherMethods
00060
00066          public bool Add(object obj)
00067          {
00068              if (obj == null) return false;
00069              if (obj is Client)
00070              {
00071                  this.ClientList.Add((Client)obj);
00072                  return true;
00073              }
00074              return false;
00075          }
00076
00082          public bool Remove(object obj)
00083          {
00084              if (obj == null) return false;
00085              var aux = (Client) obj;
00086              if (Exist(aux.ClientID))
00087              {
00088                  _clientList.Remove((Client)obj);
00089                  return true;
00090              }
00091              return false;
00092          }
00093
00099          public bool Exist(object obj)
00100          {
00101              if (obj is int)

```

```

00102         {
00103             foreach (Client client in _clientList)
00104             {
00105                 if (client.ClientID == (int)obj)
00106                 {
00107                     return true;
00108                 }
00109             }
00110         }
00111         return false;
00112     }
00113
00119     public Client GetClient(int id)
00120     {
00121         foreach (Client client in _clientList)
00122         {
00123             if (client.ClientID == id)
00124             {
00125                 return client;
00126             }
00127         }
00128         return null;
00129     }
00130 #endregion
00131
00132 #region Destructor
00136 ~Clients()
00137 {
00138 }
00139 #endregion
00140
00141 #endregion
00142 }
00143 }

```

7.15 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Interface/IListManagement.cs File Reference

Classes

- interface [trabalhoPOO_27967.Interface.IListManagement](#)

Namespaces

- namespace [trabalhoPOO_27967](#)
- namespace [trabalhoPOO_27967.Interface](#)

7.16 IListManagement.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/15/2024 04:21:43 PM</date>
00007  * <description>Defines the interface to use in List Management</description>
00008 **/
00009 using System;
00010 using System.Collections.Generic;
00011 using System.Linq;
00012 using System.Text;
00013 using System.Threading.Tasks;
00014
00015 namespace trabalhoPOO_27967.Interface
00016 {
00017     public interface IListManagement
00018     {
00019         bool Add(object obj);
00020
00021         bool Remove(object obj);
00022
00023         bool Exist(object obj);
00024     }
00025 }

```

7.17 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.18 Make.cs

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

```

00001  /*
00002  *   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *       Copyright (c) 2024 All Rights Reserved
00004  *   </copyright>
00005  *   <author>Jose Alves a27967</author>
00006  *   <date>11/6/2024 11:22:09 AM</date>
00007  *   <description></description>
00008  */
00009  using System;
00010  using System.Diagnostics.Contracts;
00011  using System.Runtime.Remoting.Messaging;
00012  using System.Text;
00013
00014  namespace trabalhoPOO_27967
00015  {
00023      public class Make
00024      {
00025          #region Attributes
00026          int _id;
00027          string _name;
00028          static int _makeCount=0;
00029          #endregion
00030
00031          #region Methods
00032
00033          #region Constructors
00034
00038              public Make()
00039              {
00040                  _id = ++_makeCount;
00041                  _name = string.Empty;
00042              }
00043
00049              public Make(string name)
00050              {
00051                  _id = ++_makeCount;
00052                  _name = name;
00053              }
00054
00055
00056
00057          #endregion
00058
00059          #region Properties
00060
00064              public int ID
00065              {
00066                  get { return _id; }
00067                  set { _id = value; }
00068              }
00069
00073              public string Name
00074              {
00075                  get { return _name; }
00076                  set { _name = value; }
00077              }

```

```

00078
00079     #endregion
00080
00081
00082
00083     #region Overrides
00088     public override string ToString()
00089     {
00090         return ("Make : " + _name);
00091     }
00092
00093
00098     public override bool Equals(object obj)
00099     {
00101         if (obj == null)
00102         {
00103             return false;
00104         }
00105
00107         Make make = obj as Make;
00108         return (this.ID == make.ID || this.Name == make.Name);
00109     }
00110
00117     public static bool operator ==(Make m1, Make m2)
00118     {
00119         return (m1.Equals(m2));
00120     }
00121
00128     public static bool operator !=(Make m1, Make m2)
00129     {
00130         return ! (m1.Equals(m2));
00131     }
00132     #endregion
00133
00134     #region OtherMethods
00135     #endregion
00136
00137     #region Destructor
00141     ~Make()
00142     {
00143     }
00144     #endregion
00145
00146     #endregion
00147 }
00148 }

```

7.19 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.20 Makes.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/14/2024 4:33:51 PM</date>

```

```

00007 * <description></description>
00008 **/
00009 using System;
00010 using System.Collections.Generic;
00011 using System.Xml.Linq;
00012 using trabalhoPOO_27967.Interface;
00013
00014 namespace trabalhoPOO_27967
00015 {
00023     public class Makes : IListManagement
00024     {
00025         #region Attributes
00026         List<Make> _makeList;
00027         #endregion
00028
00029         #region Methods
00030
00031         #region Constructors
00032
00036         public Makes()
00037         {
00038             _makeList = new List<Make>();
00039         }
00040
00045         public Makes(List<Make> m)
00046         {
00047             _makeList = m;
00048         }
00049         #endregion
00050
00051         #region Properties
00055         public List<Make> MakeList
00056         {
00057             get { return _makeList; }
00058             set { _makeList = value; }
00059         }
00060         #endregion
00061
00062
00063
00064         #region Overrides
00065         #endregion
00066
00067         #region OtherMethods
00073         public bool Add(object obj)
00074         {
00075             if (obj == null) return false;
00076             if (obj is Make) {
00077                 _makeList.Add((Make)obj);
00078                 return true;
00079             }
00080             return false;
00081         }
00082
00088         public bool Remove(object obj)
00089         {
00090             if (obj == null) return false;
00091             var aux = obj as Make;
00092             if (Exist(aux.ID))
00093             {
00094                 _makeList.Remove((Make)obj);
00095                 return true;
00096             }
00097             return false;
00098         }
00099
00105         public bool Exist(object obj)
00106         {
00107             if (obj == null) return false;
00108             if (obj is int)
00109             {
00110                 foreach (Make make in _makeList)
00111                 {
00112                     if (make.ID == (int)obj)
00113                     {
00114                         return true;
00115                     }
00116                 }
00117             }
00118             if (obj is string)
00119             {
00120                 foreach (Make make in _makeList)
00121                 {
00122                     if (make.Name == (string)obj)
00123                     {
00124                         return true;
00125                     }
00126                 }
00127             }
00128         }
00129     }
00130 }

```

```

00126         }
00127     }
00128     return false;
00129 }
00130
00131 #endregion
00132
00133 #region Destructor
00134 ~Makes()
00135 {
00136 }
00137 #endregion
00138
00139 #endregion
00140
00141 #endregion
00142 }
00143 }
00144 }

```

7.21 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs File Reference

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

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

```

00001 // <autogenerated />
00002 using System;
00003 using System.Reflection;
00004 [assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETFramework,Version=v4.7.2",
    FrameworkDisplayName = ".NET Framework 4.7.2")]

```

7.23 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.24 Product.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/2/2024 4:40:12 PM</date>
00007  * <description></description>
00008 **/
00009 using System;
00010 using System.Collections;

```



```
00011 using System.Collections.Generic;
00012 using System.Diagnostics.Contracts;
00013 using System.Text;
00014 using System.Xml.Linq;
00015
00016 namespace trabalhoPOO_27967
00017 {
00025     public class Product
00026     {
00027         #region Attributes
00028         string _reference;
00029         decimal _price;
00030         Make _make;
00031         Category _category;
00032         Warranty _warranty; //in Years
00033         int _stock;
00034         #endregion
00035
00036         #region Methods
00037
00038         #region Constructors
00039
00043         public Product()
00044         {
00045             _reference = string.Empty;
00046             _price = -1;
00047             _make = new Make();
00048             _category = new Category();
00049             _stock = 0;
00050         }
00051
00057         public Product(string reff, decimal pri, Warranty war, Make ma, Category cat)
00058         {
00059             _reference = reff;
00060             _price = pri;
00061             _make = ma;
00062             _category = cat;
00063             _warranty = war;
00064         }
00065     }
00066
00067     #endregion
00068
00069     #region Properties
00070
00074     public string Reference
00075     {
00076         get { return _reference; }
00077         set { _reference = value; }
00078     }
00079
00080
00084     public decimal Price
00085     {
00086         get { return _price; }
00087         set { _price = value; }
00088     }
00089
00093     public Make Make
00094     {
00095         get { return _make; }
00096         set { _make = value; }
00097     }
00098
00099
00103     public Category Category
00104     {
00105         get { return _category; }
00106         set { _category = value; }
00107     }
00108
00112     public int Stock
00113     {
00114         get { return _stock; }
00115         set { _stock = value; }
00116     }
00117
00118     public Warranty Warranty
00119     {
00120         get { return _warranty; }
00121         set { _warranty = value; }
00122     }
00123
00124     #endregion
00125
00126
00127     #region Overrides
```

```

00133     public override bool Equals(object obj)
00134     {
00135         if (obj == null) return false;
00136         Product product = obj as Product;
00137         if (product.Reference == _reference) return true;
00138         return false;
00139     }
00140
00147     public static bool operator ==(Product p1, Product p2)
00148     {
00149         return (p1.Equals(p2));
00150     }
00151
00158     public static bool operator !=(Product p1, Product p2)
00159     {
00160         return !(p1.Equals(p2));
00161     }
00162
00167     public override string ToString()
00168     {
00169         StringBuilder sb = new StringBuilder();
00170         sb.AppendLine($"Product ID: {_reference}");
00171         sb.AppendLine($"Price: {_price}€");
00172         sb.AppendLine(_make.ToString());
00173         sb.AppendLine(_warranty.ToString());
00174
00175         return sb.ToString();
00176     }
00177
00178     #endregion
00179
00180     #region OtherMethods
00181     #endregion
00182
00183     #region Destructor
00187     ~Product()
00188     {
00189     }
00190     #endregion
00191
00192     #endregion
00193 }
00194 }

```

7.25 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.26 Products.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/9/2024 6:34:19 PM</date>
00007  * <description></description>
00008 **/

```

```

00009 using System;
00010 using System.Collections;
00011 using System.Collections.Generic;
00012 using System.Linq;
00013 using System.Text;
00014 using trabalhoP00_27967.Interface;
00015 using static System.Net.Mime.MediaTypeNames;
00016
00017 namespace trabalhoP00_27967
00018 {
00026     public class Products : IListManagement
00027     {
00028         #region Attributes
00029         List<Product> _prods; //como faço uma propriedade para isto? Não consigo dar return do array,
00030         certo?
00031         #endregion
00032         #region Methods
00033         #region Constructors
00039         public Products()
00040         {
00041             _prods = new List<Product>();
00042         }
00043         public Products(List<Product> products)
00044         {
00045             _prods = products;
00046         }
00047         #endregion
00048         #region Properties
00061         public List<Product> Prods
00062         {
00063             get { return _prods; }
00064             set { _prods = value; }
00065         }
00066         #endregion
00067         #region Overrides
00076         public override string ToString()
00077         {
00078             StringBuilder sb = new StringBuilder();
00079             foreach (var product in _prods)
00080             {
00081                 sb.AppendLine(product.ToString());
00082             }
00083             return sb.ToString();
00084         }
00085         #endregion
00086         #region OtherMethods
00094         public decimal ValueInPosition(int p)
00095         {
00096             return this.Prods[p].Price;
00097         }
00098         public bool Add(object obj)
00099         {
00100             if (obj == null) return false;
00101             var aux=obj as Product;
00102             if (Exist(aux.Reference))
00103             {
00104                 if (obj is Product)
00105                 {
00106                     _prods.Add((Product)obj);
00107                     return true;
00108                 }
00109             }
00110             return false;
00111         }
00112         public bool Exist(object obj)
00113         {
00114             if (obj == null) return false;
00115             if (obj is string)
00116             {
00117                 foreach (Product p in _prods)
00118                 {

```

```

00131         if (p.Reference == (string)obj) return true;
00132     }
00133 }
00134 return false;
00135 }
00136
00142 public bool Remove(object obj)
00143 {
00144     if (obj == null) return false;
00145     Product p = (Product)obj;
00146     if (Exist(p.Reference))
00147     {
00148         _prods.Remove(p);
00149         return true;
00150     }
00151     return false;
00152 }
00153
00159 public Product SearchProduct(string reff)
00160 {
00161     foreach (Product p in _prods)
00162     {
00163         if(p.Reference==reff) return p;
00164     }
00165     return null;
00166 }
00167
00172 public decimal TotalPrice()
00173 {
00174     return _prods.Sum(p => p.Price);
00175 }
00176
00177
00184 public DateTime WarrantyExpirationDateForProduct(DateTime date, string reff)
00185 {
00186     Product p = SearchProduct(reff);
00187     {
00188         return (date.AddYears(p.Warranty.DurationInYears));
00189     }
00190 }
00191 #endregion
00192
00193 #region Destructor
00197 ~Products()
00198 {
00199 }
00200 #endregion
00201
00202 #endregion
00203 }
00204 }

```

7.27 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Properties/AssemblyInfo.cs File Reference

7.28 AssemblyInfo.cs

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

```

00001 using System.Reflection;
00002 using System.Runtime.CompilerServices;
00003 using System.Runtime.InteropServices;
00004
00005 // General Information about an assembly is controlled through the following
00006 // set of attributes. Change these attribute values to modify the information
00007 // associated with an assembly.
00008 [assembly: AssemblyTitle("trabalhoPOO_27967")]
00009 [assembly: AssemblyDescription("")]
00010 [assembly: AssemblyConfiguration("")]
00011 [assembly: AssemblyCompany("")]
00012 [assembly: AssemblyProduct("trabalhoPOO_27967")]
00013 [assembly: AssemblyCopyright("Copyright © 2024")]
00014 [assembly: AssemblyTrademark("")]
00015 [assembly: AssemblyCulture("")]
00016
00017 // Setting ComVisible to false makes the types in this assembly not visible
00018 // to COM components. If you need to access a type in this assembly from
00019 // COM, set the ComVisible attribute to true on that type.

```

```

00020 [assembly: ComVisible(false)]
00021
00022 // The following GUID is for the ID of the typelib if this project is exposed to COM
00023 [assembly: Guid("c7a4f6de-be70-4f08-92bd-b54d2a0111f1")]
00024
00025 // Version information for an assembly consists of the following four values:
00026 //
00027 //      Major Version
00028 //      Minor Version
00029 //      Build Number
00030 //      Revision
00031 //
00032 // You can specify all the values or you can default the Build and Revision Numbers
00033 // by using the '*' as shown below:
00034 // [assembly: AssemblyVersion("1.0.*")]
00035 [assembly: AssemblyVersion("1.0.0.0")]
00036 [assembly: AssemblyFileVersion("1.0.0.0")]

```

7.29 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.30 Sale.cs

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

```

00001 /*
00002 *   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003 *       Copyright (c) 2024 All Rights Reserved
00004 *   </copyright>
00005 *   <author>Jose Alves a27967</author>
00006 *   <date>11/6/2024 11:21:53 AM</date>
00007 *   <description></description>
00008 */
00009 using System;
00010 using System.CodeDom;
00011 using System.Linq;
00012 using System.Text;
00013
00014 namespace trabalhoPOO_27967
00015 {
00016
00017     public class Sale
00018     {
00019         #region Attributes
00020         int _id;
00021         Client _client;
00022         Products _products;
00023         decimal _totPrice;
00024         DateTime _saleDate;
00025         static int _numSales;
00026         Campaign _campaigns;
00027         #endregion
00028
00029         #region Methods
00030
00031         #region Constructors
00032         public Sale()

```

```

00046     {
00047         _products= new Products();
00048         _client= new Client();
00049         _campaigns= new Campaign();
00050     }
00051
00052     public Sale(Client client, Products products, Campaign camp)
00053     {
00054         _id = ++_numSales;
00055         _client = client;
00056         _products = products;
00057         _totPrice = TotalPrice();
00058         _saleDate = DateTime.Now;
00059     }
00060
00061     #endregion
00062
00063     #region Properties
00064     public int Id
00065     {
00066         get { return _id; }
00067         set { _id = value; }
00068     }
00069
00070     public Client Client
00071     {
00072         get { return (_client); }
00073         set { _client = value; }
00074     }
00075
00076     public Products Products
00077     {
00078         get { return _products; }
00079         set { _products = value; }
00080     }
00081
00082     public decimal TotPrice
00083     {
00084         get { return _totPrice; }
00085         set { _totPrice = value; }
00086     }
00087
00088     public DateTime SaleDate
00089     {
00090         get { return _saleDate; }
00091         set { _saleDate = value; }
00092     }
00093
00094     public Campaign Campaigns
00095     {
00096         get { return _campaigns; }
00097         set { _campaigns = value; }
00098     }
00099     #endregion
00100
00101     #region Overrides
00102     public override bool Equals(object obj)
00103     {
00104         Sale sale = obj as Sale;
00105         return (sale.Id==this.Id);
00106     }
00107
00108     public static bool operator ==(Sale s1, Sale s2)
00109     {
00110         return(s1.Equals(s2));
00111     }
00112
00113     public static bool operator !=(Sale s1, Sale s2)
00114     {
00115         return !(s1.Equals(s2));
00116     }
00117
00118     public override string ToString()
00119     {
00120         StringBuilder sb = new StringBuilder();
00121         sb.AppendLine(_products.ToString());
00122         sb.AppendLine("Total " + this.TotalPrice().ToString() + "\u20AC");
00123         return sb.ToString();
00124     }
00125     #endregion
00126
00127     #region OtherMethods

```

```

00174
00179     public decimal TotalPrice()
00180     {
00181         decimal total = 0;
00182
00183         total=_products.TotalPrice();
00184
00185         if (Campaign.VerifyApplicability(this.Campaigns))
00186         {
00187             total *= (1-this.Campaigns.Discount);
00188         }
00189
00190         return total;
00191     }
00192
00198     public bool InsertProductOnSale(Product p)
00199     {
00200         return _products.Add(p);
00201     }
00202
00208     public bool RemoveProductFromSale(Product p)
00209     {
00210         return _products.Remove(p);
00211     }
00212
00218     public bool ExistProductOnSale(Product p)
00219     {
00220         return _products.Exist(p.Reference);
00221     }
00222
00229     public DateTime WarrantyExpirationDate(string reff)
00230     {
00231         return (_products.WarrantyExpirationDateForProduct(this.SaleDate, reff));
00232     }
00233     #endregion
00234
00235     #region Destructor
00239     ~Sale()
00240     {
00241     }
00242     #endregion
00243
00244     #endregion
00245 }
00246 }

```

7.31 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.32 Sales.cs

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

```

00001 /*
00002 *   <copyright file="trabalhoPOO_27967.cs" company="IPCA">
00003 *       Copyright (c) 2024 All Rights Reserved
00004 *   </copyright>
00005 *   <author>Jose Alves a27967</author>
00006 *   <date>11/10/2024 7:42:03 PM</date>

```

```

00007 * <description></description>
00008 **/
00009 using System;
00010 using System.Collections.Generic;
00011 using trabalhoPOO_27967.Interface;
00012
00013 namespace trabalhoPOO_27967
00014 {
00022     public class Sales : IListManagement
00023     {
00024         #region Attributes
00025         static List<Sale> _salesStored;
00026         #endregion
00027
00028         #region Methods
00029
00030         #region Constructors
00031
00035         public Sales()
00036         {
00037             _salesStored = new List<Sale>();
00038         }
00039
00044         public Sales(List<Sale> sales)
00045         {
00046             _salesStored = sales;
00047         }
00048
00049         #endregion
00050
00051         #region Properties
00056         public List<Sale> SalesStored
00057         {
00058             get { return _salesStored; }
00059             set { _salesStored = value; }
00060         }
00061
00062         #endregion
00063
00064
00065
00066         #region Overrides
00067         #endregion
00068
00069         #region OtherMethods
00075         public Sale GetSale(int idSale)
00076         {
00077             foreach (Sale s in _salesStored)
00078             {
00079                 if (s.Id == idSale) return s;
00080             }
00081             return null;
00082         }
00083
00089         public bool Add(object obj)
00090         {
00091             if (obj == null) return false;
00092             if (obj is Sale)
00093             {
00094                 _salesStored.Add((Sale)obj);
00095                 return true;
00096             }
00097             return false;
00098         }
00099
00105         public bool Remove(object obj)
00106         {
00107             if (obj == null) return false;
00108             Sale aux = (Sale)obj;
00109             if (Exist(aux.Id))
00110             {
00111                 if (obj is Sale)
00112                 {
00113                     _salesStored.Remove(aux);
00114                     return true;
00115                 }
00116             }
00117             return false;
00118         }
00119
00125         public bool Exist(object obj)
00126         {
00127             if (obj == null) return false;
00128             if (obj is int)
00129             {
00130                 foreach (Sale s in _salesStored)

```



```

00131         {
00132             if (s.Id == (int)obj)
00133             {
00134                 return true;
00135             }
00136         }
00137     }
00138     return false;
00139 }
00140 #endregion
00141
00142 #region Destructor
00146 ~Sales()
00147 {
00148 }
00149 #endregion
00150
00151 #endregion
00152 }
00153 }

```

7.33 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.34 Store.cs

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

```

00001 /*
00002  * <copyright file="trabalhoPOO_27967.Store.cs" company="IPCA">
00003  *     Copyright (c) 2024 All Rights Reserved
00004  * </copyright>
00005  * <author>Jose Alves a27967</author>
00006  * <date>11/14/2024 5:01:23 PM</date>
00007  * <description></description>
00008  */
00009 using System;
00010
00011 namespace trabalhoPOO_27967.Store
00012 {
00020     public class Store
00021     {
00022         #region Attributes
00023         static Clients _clientList;
00024         static Products _prodList;
00025         static Sales _saleList;
00026         static Makes _makeList;
00027         static Categories _catList;
00028         static Warranties _warrantList;
00029         #endregion
00030
00031         #region Methods
00032
00033         #region Constructors
00034
00038         public Store()
00039         {

```

```

00040         _clientList = new Clients();
00041         _prodList = new Products();
00042         _saleList = new Sales();
00043         _makeList = new Makes();
00044         _catList = new Categories();
00045         _warrantList = new Warranties();
00046     }
00047
00057 public Store(Clients cl, Products p, Sales s, Makes m, Categories c, Warranties w)
00058 {
00059     _clientList=cl;
00060     _prodList=p;
00061     _saleList=s;
00062     _makeList=m;
00063     _catList=c;
00064     _warrantList=w;
00065 }
00066
00067 #endregion
00068
00069 #region Properties
00073 public Clients ClientList
00074 {
00075     get { return _clientList; }
00076     set { _clientList = value; }
00077 }
00078
00082 public Products ProdList
00083 {
00084     get { return _prodList; }
00085     set { _prodList = value; }
00086 }
00087
00091 public Sales SaleList
00092 {
00093     get { return _saleList; }
00094     set { _saleList = value; }
00095 }
00096
00100 public Makes MakeList
00101 {
00102     get { return _makeList; }
00103     set { _makeList = value; }
00104 }
00105
00109 public Categories CatList
00110 {
00111     get { return _catList; }
00112     set { _catList = value; }
00113 }
00114
00118 public Warranties WarrantList
00119 {
00120     get { return _warrantList; }
00121     set { _warrantList = value; }
00122 }
00123
00124
00125 #endregion
00126
00127
00128
00129 #region Overrides
00130 #endregion
00131
00132 #region OtherMethods
00133 #endregion
00134
00135 #region Destructor
00139 ~Store()
00140 {
00141 }
00142 #endregion
00143
00144 #endregion
00145 }
00146 }

```

7.35 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.36 Warranties.cs

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

```

00001  /*
00002  *   <copyright file="trabalhoPOO_27967.Category.cs" company="IPCA">
00003  *       Copyright (c) 2024 All Rights Reserved
00004  *   </copyright>
00005  *   <author>Jose Alves a27967</author>
00006  *   <date>11/14/2024 4:20:11 PM</date>
00007  *   <description></description>
00008  */
00009  using System;
00010  using System.Collections.Generic;
00011  using trabalhoPOO_27967.Interface;
00012
00013  namespace trabalhoPOO_27967
00014  {
00022      public class Warranties: IListManagement
00023      {
00024          #region Attributes
00025          List<Warranty> _warrants;
00026          #endregion
00027
00028          #region Methods
00029
00030          #region Constructors
00031
00035          public Warranties()
00036          {
00037              _warrants = new List<Warranty>();
00038          }
00039
00044          public Warranties(List<Warranty> warrants)
00045          {
00046              _warrants = warrants;
00047          }
00048
00049
00050
00051          #endregion
00052
00053          #region Properties
00057          public List<Warranty> Warrants
00058          {
00059              get { return _warrants; }
00060              set { _warrants = value; }
00061          }
00062          #endregion
00063
00064
00065
00066          #region Overrides
00067          #endregion
00068
00069          #region OtherMethods
00070          // <summary>
00075          public bool Add(object obj)
00076          {
00077              if (obj == null) return false;

```

```

00078         if(obj is Warranty)
00079         {
00080             _warrants.Add((Warranty)obj);
00081             return true;
00082         }
00083         return false;
00084     }
00085
00091     public bool Remove(object obj)
00092     {
00093         if (obj == null) return false;
00094         var aux = obj as Warranty;
00095         if (Exist(aux.ProdID))
00096         {
00097             if (obj is Warranty)
00098             {
00099                 _warrants.Remove((Warranty)obj);
00100                 return true;
00101             }
00102         }
00103         return false;
00104     }
00105
00111     public bool Exist(object obj)
00112     {
00113         if (obj == null) return false;
00114         if (obj is string)
00115         {
00116             foreach (Warranty w in _warrants)
00117             {
00118                 if (w.ProdID == (string)obj)
00119                 {
00120                     return true;
00121                 }
00122             }
00123         }
00124         return false;
00125     }
00126 #endregion
00127
00128 #region Destructor
00132 ~Warranties()
00133 {
00134 }
00135 #endregion
00136
00137 #endregion
00138 }
00139 }

```

7.37 D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/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](#)

7.38 Warranty.cs

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

```

00001 /*
00002 * <copyright file="trabalhoPOO_27967.cs" company="IPCA">

```

```
00003 *      Copyright (c) 2024 All Rights Reserved
00004 *      </copyright>
00005 *      <author>Jose Alves a27967</author>
00006 *      <date>11/13/2024 4:17:18 PM</date>
00007 *      <description></description>
00008 **/
00009 using System;
00010 using System.Diagnostics.Contracts;
00011 using System.Text;
00012 using System.Xml.Linq;
00013
00014 namespace trabalhoPOO_27967
00015 {
00023     public class Warranty
00024     {
00025         #region Attributes
00026         string _prodID;
00027         int _durationInYears;
00028         string _conditions;
00029         #endregion
00030
00031         #region Methods
00032
00033         #region Constructors
00038         public Warranty()
00039         {
00040             _prodID = string.Empty;
00041             _durationInYears = -1;
00042             _conditions = string.Empty;
00043         }
00044
00051         public Warranty(string prodID, int durationInYears, string conditions)
00052         {
00053             _prodID = prodID;
00054             _durationInYears = durationInYears;
00055             _conditions = conditions;
00056         }
00057
00058         #endregion
00059
00060         #region Properties
00063
00067         public string ProdID
00068         {
00069             get { return _prodID; }
00070             set { _prodID = value; }
00071         }
00072
00076         public int DurationInYears
00077         {
00078             get { return _durationInYears; }
00079             set { _durationInYears = value; }
00080         }
00081
00085         public string Conditions
00086         {
00087             get { return _conditions; }
00088             set { _conditions = value; }
00089         }
00090         #endregion
00091
00092         #region Overrides
00094         public override string ToString()
00100         {
00101             StringBuilder sb = new StringBuilder();
00102             sb.AppendLine($"Warranty Data: ");
00103             sb.AppendLine($"Duration: {_durationInYears} years");
00104             sb.AppendLine($"Terms: {_conditions}");
00105
00106             return sb.ToString();
00107         }
00108
00114         public override bool Equals(object obj)
00115         {
00117             if (obj == null)
00118             {
00119                 return false;
00120             }
00121
00123             Warranty war = obj as Warranty;
00124             return (this.ProdID == war.ProdID && this.Conditions == war.Conditions);
00125         }
00125     }
```

```
00126
00133     public static bool operator ==(Warranty w1, Warranty w2)
00134     {
00135         return (w1.Equals(w2));
00136     }
00137
00144     public static bool operator !=(Warranty w1, Warranty w2)
00145     {
00146         return !(w1.Equals(w2));
00147     }
00148     #endregion
00149
00150     #region OtherMethods
00157     public DateTime ExpirationDate(Sale s, string reff)
00158     {
00159         Product p = s.Products.SearchProduct(reff);
00160         return (s.SaleDate.AddYears(_durationInYears));
00161     }
00162     #endregion
00163
00164     #region Destructor
00168     ~Warranty()
00169     {
00170     }
00171     #endregion
00172
00173     #endregion
00174 }
00175 }
```

Index

Add

trabalhoPOO_27967.Categories, [18](#)
trabalhoPOO_27967.Clients, [28](#)
trabalhoPOO_27967.Interface.IListManagement, [30](#)
trabalhoPOO_27967.Makes, [35](#)
trabalhoPOO_27967.Products, [42](#)
trabalhoPOO_27967.Sales, [52](#)
trabalhoPOO_27967.Warranties, [57](#)

Campaign

trabalhoPOO_27967.Campaign, [12](#)

CampaignCount

trabalhoPOO_27967.Campaign, [15](#)

Campaigns

trabalhoPOO_27967.Sale, [50](#)

Categories

trabalhoPOO_27967.Categories, [17](#)

Category

trabalhoPOO_27967.Category, [20](#)
trabalhoPOO_27967.Product, [39](#)

CatList

trabalhoPOO_27967.Store.Store, [55](#)

Cats

trabalhoPOO_27967.Categories, [19](#)

Client

trabalhoPOO_27967.Client, [23](#)
trabalhoPOO_27967.Sale, [50](#)

ClientCount

trabalhoPOO_27967.Client, [26](#)

ClientID

trabalhoPOO_27967.Client, [26](#)

ClientLlist

trabalhoPOO_27967.Store.Store, [55](#)

ClientList

trabalhoPOO_27967.Clients, [29](#)

Clients

trabalhoPOO_27967.Clients, [28](#)

Conditions

trabalhoPOO_27967.Warranty, [62](#)

Contact

trabalhoPOO_27967.Client, [26](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/BestSale.cs, [63](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaign.cs, [64](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Campaign/Campaigns.cs, [66](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Categories.cs, [67, 68](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Category/Category.cs, [69](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Client.cs, [70, 71](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Client/Clients.cs, [72, 73](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Interface/IListManagement.cs, [74](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Make.cs, [75](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Make/Makes.cs, [76](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/obj/Debug/.NETFramework, [78](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Product.cs, [78](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Product/Products.cs, [80](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Properties/AssemblyInfo.cs, [82](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sale.cs, [83](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Sale/Sales.cs, [85](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Store/Store.cs, [87](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranties.cs, [89](#)

D:/Aulas/POO/TP_POO_27967/trabalhoPOO_27967/Warranty/Warranty.cs, [90](#)

Discount

trabalhoPOO_27967.Campaign, [15](#)

DurationInYears

trabalhoPOO_27967.Warranty, [62](#)

EndDate

trabalhoPOO_27967.Campaign, [15](#)

Equals

trabalhoPOO_27967.Campaign, [13](#)

trabalhoPOO_27967.Category, [21](#)

trabalhoPOO_27967.Client, [24](#)

trabalhoPOO_27967.Make, [32](#)

trabalhoPOO_27967.Product, [38](#)

trabalhoPOO_27967.Sale, [47](#)

trabalhoPOO_27967.Warranty, [60](#)

trabalhoPOO_27967.Categories, [18](#)

trabalhoPOO_27967.Clients, [28](#)

- trabalhoPOO_27967.Interface.IListManagement, 30
- trabalhoPOO_27967.Makes, 35
- trabalhoPOO_27967.Products, 42
- trabalhoPOO_27967.Sales, 52
- trabalhoPOO_27967.Warranties, 58
- ExistProductOnSale
 - trabalhoPOO_27967.Sale, 47
- ExpirationDate
 - trabalhoPOO_27967.Warranty, 60
- GetClient
 - trabalhoPOO_27967.Clients, 28
- GetSale
 - trabalhoPOO_27967.Sales, 52
- ID
 - trabalhoPOO_27967.Make, 33
- Id
 - trabalhoPOO_27967.Campaign, 16
 - trabalhoPOO_27967.Category, 22
 - trabalhoPOO_27967.Sale, 50
- InsertProductOnSale
 - trabalhoPOO_27967.Sale, 47
- Make
 - trabalhoPOO_27967.Make, 31
 - trabalhoPOO_27967.Product, 39
- MakeList
 - trabalhoPOO_27967.Makes, 36
 - trabalhoPOO_27967.Store.Store, 55
- Makes
 - trabalhoPOO_27967.Makes, 34
- Name
 - trabalhoPOO_27967.Campaign, 16
 - trabalhoPOO_27967.Category, 22
 - trabalhoPOO_27967.Client, 26
 - trabalhoPOO_27967.Make, 33
- operator!=
 - trabalhoPOO_27967.Campaign, 13
 - trabalhoPOO_27967.Category, 21
 - trabalhoPOO_27967.Client, 24
 - trabalhoPOO_27967.Make, 32
 - trabalhoPOO_27967.Product, 38
 - trabalhoPOO_27967.Sale, 48
 - trabalhoPOO_27967.Warranty, 61
- operator==
 - trabalhoPOO_27967.Campaign, 13
 - trabalhoPOO_27967.Category, 21
 - trabalhoPOO_27967.Client, 24
 - trabalhoPOO_27967.Make, 32
 - trabalhoPOO_27967.Product, 38
 - trabalhoPOO_27967.Sale, 48
 - trabalhoPOO_27967.Warranty, 61
- Price
 - trabalhoPOO_27967.Product, 39
- ProdID
 - trabalhoPOO_27967.Warranty, 62
- ProdList
 - trabalhoPOO_27967.Store.Store, 55
- Prods
 - trabalhoPOO_27967.Products, 45
- Product
 - trabalhoPOO_27967.Product, 37
- Products
 - trabalhoPOO_27967.Products, 41
 - trabalhoPOO_27967.Sale, 50
- Reference
 - trabalhoPOO_27967.Product, 40
- Remove
 - trabalhoPOO_27967.Categories, 18
 - trabalhoPOO_27967.Clients, 29
 - trabalhoPOO_27967.Interface.IListManagement, 30
 - trabalhoPOO_27967.Makes, 35
 - trabalhoPOO_27967.Products, 42
 - trabalhoPOO_27967.Sales, 53
 - trabalhoPOO_27967.Warranties, 58
- RemoveProductFromSale
 - trabalhoPOO_27967.Sale, 48
- Sale
 - trabalhoPOO_27967.Sale, 46
- SaleDate
 - trabalhoPOO_27967.Sale, 50
- SaleList
 - trabalhoPOO_27967.Store.Store, 55
- Sales
 - trabalhoPOO_27967.Sales, 51
- SalesStored
 - trabalhoPOO_27967.Sales, 53
- SearchProduct
 - trabalhoPOO_27967.Products, 43
- StartDate
 - trabalhoPOO_27967.Campaign, 16
- Stock
 - trabalhoPOO_27967.Product, 40
- Store
 - trabalhoPOO_27967.Store.Store, 54
- ToString
 - trabalhoPOO_27967.Client, 26
 - trabalhoPOO_27967.Make, 33
 - trabalhoPOO_27967.Product, 39
 - trabalhoPOO_27967.Products, 43
 - trabalhoPOO_27967.Sale, 49
 - trabalhoPOO_27967.Warranty, 61
- TotalPrice
 - trabalhoPOO_27967.Products, 43
 - trabalhoPOO_27967.Sale, 49
- TotPrice
 - trabalhoPOO_27967.Sale, 50
- trabalhoPOO_27967, 9
- trabalhoPOO_27967.BestSale, 11
- trabalhoPOO_27967.Campaign, 11

- Campaign, [12](#)
- CampaignCount, [15](#)
- Discount, [15](#)
- EndDate, [15](#)
- Equals, [13](#)
- Id, [16](#)
- Name, [16](#)
- operator!=, [13](#)
- operator==, [13](#)
- StartDate, [16](#)
- VerifyApplicability, [15](#)
- trabalhoPOO_27967.Campaigns, [16](#)
- trabalhoPOO_27967.Categories, [17](#)
 - Add, [18](#)
 - Categories, [17](#)
 - Cats, [19](#)
 - Exist, [18](#)
 - Remove, [18](#)
- trabalhoPOO_27967.Category, [19](#)
 - Category, [20](#)
 - Equals, [21](#)
 - Id, [22](#)
 - Name, [22](#)
 - operator!=, [21](#)
 - operator==, [21](#)
- trabalhoPOO_27967.Client, [22](#)
 - Client, [23](#)
 - ClientCount, [26](#)
 - ClientID, [26](#)
 - Contact, [26](#)
 - Equals, [24](#)
 - Name, [26](#)
 - operator!=, [24](#)
 - operator==, [24](#)
 - ToString, [26](#)
- trabalhoPOO_27967.Clients, [27](#)
 - Add, [28](#)
 - ClientList, [29](#)
 - Clients, [28](#)
 - Exist, [28](#)
 - GetClient, [28](#)
 - Remove, [29](#)
- trabalhoPOO_27967.Interface, [10](#)
- trabalhoPOO_27967.Interface.IListManagement, [30](#)
 - Add, [30](#)
 - Exist, [30](#)
 - Remove, [30](#)
- trabalhoPOO_27967.Make, [31](#)
 - Equals, [32](#)
 - ID, [33](#)
 - Make, [31](#)
 - Name, [33](#)
 - operator!=, [32](#)
 - operator==, [32](#)
 - ToString, [33](#)
- trabalhoPOO_27967.Makes, [34](#)
 - Add, [35](#)
 - Exist, [35](#)
 - MakeList, [36](#)
 - Makes, [34](#)
 - Remove, [35](#)
- trabalhoPOO_27967.Product, [36](#)
 - Category, [39](#)
 - Equals, [38](#)
 - Make, [39](#)
 - operator!=, [38](#)
 - operator==, [38](#)
 - Price, [39](#)
 - Product, [37](#)
 - Reference, [40](#)
 - Stock, [40](#)
 - ToString, [39](#)
 - Warranty, [40](#)
- trabalhoPOO_27967.Products, [40](#)
 - Add, [42](#)
 - Exist, [42](#)
 - Prods, [45](#)
 - Products, [41](#)
 - Remove, [42](#)
 - SearchProduct, [43](#)
 - ToString, [43](#)
 - TotalPrice, [43](#)
 - ValueInPosition, [44](#)
 - WarrantyExpirationDateForProduct, [44](#)
- trabalhoPOO_27967.Sale, [45](#)
 - Campaigns, [50](#)
 - Client, [50](#)
 - Equals, [47](#)
 - ExistProductOnSale, [47](#)
 - Id, [50](#)
 - InsertProductOnSale, [47](#)
 - operator!=, [48](#)
 - operator==, [48](#)
 - Products, [50](#)
 - RemoveProductFromSale, [48](#)
 - Sale, [46](#)
 - SaleDate, [50](#)
 - ToString, [49](#)
 - TotalPrice, [49](#)
 - TotPrice, [50](#)
 - WarrantyExpirationDate, [49](#)
- trabalhoPOO_27967.Sales, [51](#)
 - Add, [52](#)
 - Exist, [52](#)
 - GetSale, [52](#)
 - Remove, [53](#)
 - Sales, [51](#)
 - SalesStored, [53](#)
- trabalhoPOO_27967.Store, [10](#)
- trabalhoPOO_27967.Store.Store, [54](#)
 - CatList, [55](#)
 - ClientList, [55](#)
 - MakeList, [55](#)
 - ProdList, [55](#)
 - SaleList, [55](#)
 - Store, [54](#)

- WarrantList, [56](#)
- trabalhoPOO_27967.Warranties, [56](#)
 - Add, [57](#)
 - Exist, [58](#)
 - Remove, [58](#)
 - Warranties, [57](#)
 - Warrants, [58](#)
- trabalhoPOO_27967.Warranty, [59](#)
 - Conditions, [62](#)
 - DurationInYears, [62](#)
 - Equals, [60](#)
 - ExpirationDate, [60](#)
 - operator!=, [61](#)
 - operator==, [61](#)
 - ProdID, [62](#)
 - ToString, [61](#)
 - Warranty, [59](#)
- ValueInPosition
 - trabalhoPOO_27967.Products, [44](#)
- VerifyApplicability
 - trabalhoPOO_27967.Campaign, [15](#)
- Warranties
 - trabalhoPOO_27967.Warranties, [57](#)
- WarrantList
 - trabalhoPOO_27967.Store.Store, [56](#)
- Warrants
 - trabalhoPOO_27967.Warranties, [58](#)
- Warranty
 - trabalhoPOO_27967.Product, [40](#)
 - trabalhoPOO_27967.Warranty, [59](#)
- WarrantyExpirationDate
 - trabalhoPOO_27967.Sale, [49](#)
- WarrantyExpirationDateForProduct
 - trabalhoPOO_27967.Products, [44](#)