

LESI_TP_POO_27967

v 1.0

Generated by Doxygen 1.9.8

1 Namespace Index	1
1.1 Package List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 trabalhoPOO_27967 Namespace Reference	7
4.2 trabalhoPOO_27967.Store Namespace Reference	8
5 Class Documentation	9
5.1 trabalhoPOO_27967.BestSale Class Reference	9
5.1.1 Detailed Description	9
5.2 trabalhoPOO_27967.Campaign Class Reference	9
5.2.1 Detailed Description	10
5.2.2 Constructor & Destructor Documentation	10
5.2.2.1 Campaign() [1/2]	10
5.2.2.2 Campaign() [2/2]	10
5.2.3 Member Function Documentation	11
5.2.3.1 Equals()	11
5.2.3.2 operator!=(())	11
5.2.3.3 operator==(())	11
5.2.3.4 VerifyApplicability()	13
5.2.4 Property Documentation	13
5.2.4.1 CampaignCount	13
5.2.4.2 Discount	13
5.2.4.3 EndDate	14
5.2.4.4 Id	14
5.2.4.5 Name	14
5.2.4.6 StartDate	14
5.3 trabalhoPOO_27967.Campaigns Class Reference	14
5.3.1 Detailed Description	15
5.3.2 Constructor & Destructor Documentation	15
5.3.2.1 Campaigns() [1/2]	15
5.3.2.2 Campaigns() [2/2]	15
5.3.3 Member Function Documentation	16
5.3.3.1 AddCampaign()	16
5.3.3.2 ExistCampaign()	16
5.3.3.3 RemoveCampaign()	16
5.3.4 Property Documentation	17
5.3.4.1 Camps	17

5.4 trabalhoPOO_27967.Categories Class Reference	17
5.4.1 Detailed Description	18
5.4.2 Constructor & Destructor Documentation	18
5.4.2.1 Categories() [1/2]	18
5.4.2.2 Categories() [2/2]	18
5.4.3 Member Function Documentation	18
5.4.3.1 AddCategory()	18
5.4.3.2 ExistCategory() [1/2]	19
5.4.3.3 ExistCategory() [2/2]	19
5.4.3.4 RemoveCategory()	19
5.4.4 Property Documentation	20
5.4.4.1 Cats	20
5.5 trabalhoPOO_27967.Category Class Reference	20
5.5.1 Detailed Description	21
5.5.2 Constructor & Destructor Documentation	21
5.5.2.1 Category() [1/2]	21
5.5.2.2 Category() [2/2]	21
5.5.3 Member Function Documentation	21
5.5.3.1 Equals()	21
5.5.3.2 operator!=(())	22
5.5.3.3 operator==(())	22
5.5.4 Property Documentation	22
5.5.4.1 Id	22
5.5.4.2 Name	23
5.6 trabalhoPOO_27967.Client Class Reference	23
5.6.1 Detailed Description	24
5.6.2 Constructor & Destructor Documentation	24
5.6.2.1 Client() [1/2]	24
5.6.2.2 Client() [2/2]	24
5.6.3 Member Function Documentation	24
5.6.3.1 Equals()	24
5.6.3.2 operator!=(())	25
5.6.3.3 operator==(())	25
5.6.3.4 ToString()	26
5.6.4 Property Documentation	26
5.6.4.1 ClientCount	26
5.6.4.2 ClientID	26
5.6.4.3 Contact	26
5.6.4.4 Name	26
5.7 trabalhoPOO_27967.Clients Class Reference	27
5.7.1 Detailed Description	27
5.7.2 Constructor & Destructor Documentation	27

5.7.2.1 Clients()	27
5.7.3 Member Function Documentation	27
5.7.3.1 AddClient()	27
5.7.3.2 ExistClient()	28
5.7.3.3 GetClient()	28
5.7.3.4 RemoveClient()	28
5.7.4 Property Documentation	29
5.7.4.1 ClientList	29
5.8 trabalhoPOO_27967.Make Class Reference	29
5.8.1 Detailed Description	30
5.8.2 Constructor & Destructor Documentation	30
5.8.2.1 Make() [1/2]	30
5.8.2.2 Make() [2/2]	30
5.8.3 Member Function Documentation	30
5.8.3.1 Equals()	30
5.8.3.2 operator!=(())	31
5.8.3.3 operator==(())	31
5.8.3.4 ToString()	32
5.8.4 Property Documentation	32
5.8.4.1 ID	32
5.8.4.2 Name	32
5.9 trabalhoPOO_27967.Makes Class Reference	32
5.9.1 Detailed Description	33
5.9.2 Constructor & Destructor Documentation	33
5.9.2.1 Makes() [1/2]	33
5.9.2.2 Makes() [2/2]	33
5.9.3 Member Function Documentation	34
5.9.3.1 AddMake()	34
5.9.3.2 ExistMake() [1/2]	34
5.9.3.3 ExistMake() [2/2]	34
5.9.3.4 RemoveMake()	35
5.9.4 Property Documentation	35
5.9.4.1 MakeList	35
5.10 trabalhoPOO_27967.Product Class Reference	35
5.10.1 Detailed Description	36
5.10.2 Constructor & Destructor Documentation	36
5.10.2.1 Product() [1/2]	36
5.10.2.2 Product() [2/2]	36
5.10.3 Member Function Documentation	37
5.10.3.1 Equals()	37
5.10.3.2 operator!=(())	37
5.10.3.3 operator==(())	38

5.10.3.4 ToString()	38
5.10.4 Property Documentation	38
5.10.4.1 Category	38
5.10.4.2 Make	38
5.10.4.3 Price	39
5.10.4.4 Reference	39
5.10.4.5 Stock	39
5.10.4.6 Warranty	39
5.11 trabalhoPOO_27967.Products Class Reference	39
5.11.1 Detailed Description	40
5.11.2 Constructor & Destructor Documentation	40
5.11.2.1 Products() [1/2]	40
5.11.2.2 Products() [2/2]	40
5.11.3 Member Function Documentation	41
5.11.3.1 ExistProduct()	41
5.11.3.2 InsertProduct()	41
5.11.3.3 RemoveProduct()	41
5.11.3.4 SearchProduct()	42
5.11.3.5 ToString()	42
5.11.3.6 TotalPrice()	42
5.11.3.7 ValueInPosition()	43
5.11.3.8 WarratyExpirationDateForProduct()	43
5.11.4 Property Documentation	43
5.11.4.1 Prods	43
5.12 trabalhoPOO_27967.Sale Class Reference	44
5.12.1 Detailed Description	45
5.12.2 Constructor & Destructor Documentation	45
5.12.2.1 Sale() [1/2]	45
5.12.2.2 Sale() [2/2]	45
5.12.3 Member Function Documentation	45
5.12.3.1 Equals()	45
5.12.3.2 ExistProductOnSale()	46
5.12.3.3 InsertProductOnSale()	46
5.12.3.4 operator!=(())	46
5.12.3.5 operator==(())	47
5.12.3.6 RemoveProductFromSale()	47
5.12.3.7 ToString()	48
5.12.3.8 TotalPrice()	48
5.12.3.9 WarrantyExpirationDate()	48
5.12.4 Property Documentation	48
5.12.4.1 Campaigns	48
5.12.4.2 Client	49

5.12.4.3 Id	49
5.12.4.4 Products	49
5.12.4.5 SaleDate	49
5.12.4.6 TotPrice	49
5.13 trabalhoPOO_27967.Sales Class Reference	49
5.13.1 Detailed Description	50
5.13.2 Constructor & Destructor Documentation	50
5.13.2.1 Sales() [1/2]	50
5.13.2.2 Sales() [2/2]	50
5.13.3 Member Function Documentation	51
5.13.3.1 AddSale()	51
5.13.3.2 GetSale()	51
5.13.4 Property Documentation	51
5.13.4.1 SalesStored	51
5.14 trabalhoPOO_27967.Store.Store Class Reference	52
5.14.1 Detailed Description	52
5.14.2 Constructor & Destructor Documentation	52
5.14.2.1 Store() [1/2]	52
5.14.2.2 Store() [2/2]	52
5.14.3 Property Documentation	53
5.14.3.1 CatList	53
5.14.3.2 ClientList	53
5.14.3.3 MakeList	53
5.14.3.4 ProdList	53
5.14.3.5 SaleList	54
5.14.3.6 WarrantList	54
5.15 trabalhoPOO_27967.Warranties Class Reference	54
5.15.1 Detailed Description	54
5.15.2 Constructor & Destructor Documentation	55
5.15.2.1 Warranties() [1/2]	55
5.15.2.2 Warranties() [2/2]	55
5.15.3 Member Function Documentation	55
5.15.3.1 AddWarranty()	55
5.15.3.2 ExistWarranty()	55
5.15.3.3 RemoveWarranty()	56
5.15.4 Property Documentation	56
5.15.4.1 Warrants	56
5.16 trabalhoPOO_27967.Warranty Class Reference	56
5.16.1 Detailed Description	57
5.16.2 Constructor & Destructor Documentation	57
5.16.2.1 Warranty() [1/2]	57
5.16.2.2 Warranty() [2/2]	57

5.16.3 Member Function Documentation	58
5.16.3.1 Equals()	58
5.16.3.2 ExpirationDate()	58
5.16.3.3 operator!=()	59
5.16.3.4 operator==()	59
5.16.3.5 ToString()	59
5.16.4 Property Documentation	60
5.16.4.1 Conditions	60
5.16.4.2 DurationInYears	60
5.16.4.3 ProdID	60
6 File Documentation	61
6.1 BestSale.cs File Reference	61
6.2 BestSale.cs	61
6.3 Campaign/Campaign.cs File Reference	62
6.4 Campaign.cs	62
6.5 Campaign/Campaigns.cs File Reference	64
6.6 Campaigns.cs	64
6.7 Category/Categories.cs File Reference	65
6.8 Categories.cs	65
6.9 Category/Category.cs File Reference	67
6.10 Category.cs	67
6.11 Client/Client.cs File Reference	68
6.12 Client.cs	68
6.13 Client/Clients.cs File Reference	70
6.14 Clients.cs	70
6.15 Make/Make.cs File Reference	72
6.16 Make.cs	72
6.17 Make/Makes.cs File Reference	73
6.18 Makes.cs	73
6.19 obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs File Reference	75
6.20 .NETFramework,Version=v4.7.2.AssemblyAttributes.cs	75
6.21 Product/Product.cs File Reference	75
6.22 Product.cs	75
6.23 Product/Products.cs File Reference	77
6.24 Products.cs	77
6.25 Properties/AssemblyInfo.cs File Reference	79
6.26 AssemblyInfo.cs	79
6.27 Sale/Sale.cs File Reference	79
6.28 Sale.cs	80
6.29 Sale/Sales.cs File Reference	82
6.30 Sales.cs	82

6.31 Store/Store.cs File Reference	83
6.32 Store.cs	83
6.33 Warranty/Warranties.cs File Reference	85
6.34 Warranties.cs	85
6.35 Warranty/Warranty.cs File Reference	86
6.36 Warranty.cs	86
Index	89

Chapter 1

Namespace Index

1.1 Package List

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

trabalhoPOO_27967	7
trabalhoPOO_27967.Store	8

Chapter 2

Class Index

2.1 Class List

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

trabalhoPOO_27967.BestSale	9
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	9
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	14
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	20
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	23
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.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	29
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	32
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	35
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	39
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	44

[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 49

[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 52

[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 54

[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 56

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

BestSale.cs	61
Campaign/ Campaign.cs	62
Campaign/ Campaigns.cs	64
Category/ Categories.cs	65
Category/ Category.cs	67
Client/ Client.cs	68
Client/ Clients.cs	70
Make/ Make.cs	72
Make/ Makes.cs	73
obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs	75
Product/ Product.cs	75
Product/ Products.cs	77
Properties/ AssemblyInfo.cs	79
Sale/ Sale.cs	79
Sale/ Sales.cs	82
Store/ Store.cs	83
Warranty/ Warranties.cs	85
Warranty/ Warranty.cs	86

Chapter 4

Namespace Documentation

4.1 trabalhoPOO_27967 Namespace Reference

Namespaces

- namespace [Store](#)

Classes

- class [BestSale](#)
- class [Campaign](#)
Purpose: Definition of Campaign and methods to deal with Campaign operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:21:43 AM.
- class [Campaigns](#)
Purpose: This file has the definition and methods to work with the plurality of Campaign. Created by: Jose Alves a27967 Created on: 11/14/2024 3:57:04 PM.
- class [Categories](#)
Purpose: This file has the definition and methods to work with the plurality of Category. Created by: Jose Alves a27967 Created on: 11/14/2024 4:45:58 PM.
- class [Category](#)
Purpose: Definition of Category and methods to deal with Category operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:20:47 AM.
- class [Client](#)
Purpose: Definition of Client and methods to deal with Client operations. Created by: Jose Alves a27967 Created on: 10/29/2024 4:23:56 PM.
- class [Clients](#)
Purpose: Class with the definition and methods to manage a list of clients. Created by: Jose Alves a27967 Created on: 11/12/2024 9:25:28 PM.
- class [Make](#)
Purpose: Definition of Make and methods to deal with Make operations. Created by: Jose Alves a27967 Created on: 11/6/2024 11:22:09 AM.
- class [Makes](#)
Purpose: This file has the definition and methods to work with the plurality of Make. Created by: Jose Alves a27967 Created on: 11/14/2024 4:33:51 PM.
- class [Product](#)
Purpose: Definition of product and methods to deal with product operations. Created by: Jose Alves a27967 Created on: 11/2/2024 4:40:12 PM.

- class [Products](#)

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

- class [Sale](#)

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

- class [Sales](#)

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

- class [Warranties](#)

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

- class [Warranty](#)

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

4.2 trabalhoPOO_27967.Store Namespace Reference

Classes

- class [Store](#)

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

Chapter 5

Class Documentation

5.1 trabalhoPOO_27967.BestSale Class Reference

5.1.1 Detailed Description

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

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

- [BestSale.cs](#)

5.2 trabalhoPOO_27967.Campaign Class Reference

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

Public Member Functions

- [Campaign](#) ()
The default Constructor.
- [Campaign](#) (string name, decimal discount, DateTime startDate, DateTime endDate)
Constructor used to create a campaign when a name, a decimal value of discount, start and ending dates are given.
- override bool [Equals](#) (object obj)
Redefine the Equals operator to verify if a campaign matches the other.

Static Public Member Functions

- static bool [operator==](#) ([Campaign](#) camp1, [Campaign](#) camp2)
Redefinition of the == operator.
- static bool [operator!=](#) ([Campaign](#) camp1, [Campaign](#) camp2)
Redefinition of the != operator.
- static bool [VerifyApplicability](#) ([Campaign](#) camp)
Method to verify if a 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.

5.2.1 Detailed Description

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

Definition at line 20 of file [Campaign.cs](#).

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Campaign() [1/2]

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

The default Constructor.

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

5.2.2.2 Campaign() [2/2]

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

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

Parameters

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

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

5.2.3 Member Function Documentation

5.2.3.1 Equals()

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

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

Parameters

<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](#).

5.2.3.2 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](#).

5.2.3.3 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](#).

5.2.3.4 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](#).

5.2.4 Property Documentation**5.2.4.1 CampaignCount**

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

Property used to get and set the Campaign Counter.

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

5.2.4.2 Discount

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

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

Definition at line 92 of file [Campaign.cs](#).

5.2.4.3 EndDate

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

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

Definition at line 110 of file [Campaign.cs](#).

5.2.4.4 Id

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

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

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

5.2.4.5 Name

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

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

Definition at line 83 of file [Campaign.cs](#).

5.2.4.6 StartDate

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

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

Definition at line 101 of file [Campaign.cs](#).

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

- Campaign/[Campaign.cs](#)

5.3 trabalhoPOO_27967.Campaigns Class Reference

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

Public Member Functions

- [Campaigns](#) ()
The default Constructor.
- [Campaigns](#) (List< [Campaign](#) > p)
The constructor to use when a list of Campaigns is given.
- bool [AddCampaign](#) ([Campaign](#) c)
Method used to add a campaign to a list of campaigns.
- bool [RemoveCampaign](#) ([Campaign](#) camp)
Method used to remove a campaign from a list of campaigns.
- bool [ExistCampaign](#) (int id)
Method used to confirm if a campaign exists on a list of campaigns.

Properties

- List< [Campaign](#) > [Camps](#) [get, set]
Property used to get and set a Campaign list.

5.3.1 Detailed Description

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

Definition at line 22 of file [Campaigns.cs](#).

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Campaigns() [1/2]

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

The default Constructor.

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

5.3.2.2 Campaigns() [2/2]

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

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

Parameters

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

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

5.3.3 Member Function Documentation

5.3.3.1 AddCampaign()

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

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

Parameters

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

Returns

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

5.3.3.2 ExistCampaign()

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

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

Parameters

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

Returns

Definition at line 103 of file [Campaigns.cs](#).

5.3.3.3 RemoveCampaign()

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

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

Parameters

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

Returns

Definition at line 85 of file [Campaigns.cs](#).

5.3.4 Property Documentation

5.3.4.1 Camps

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

Property used to get and set a Campaign list.

Definition at line 54 of file [Campaigns.cs](#).

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

- Campaign/[Campaigns.cs](#)

5.4 trabalhoPOO_27967.Categories Class Reference

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

Public Member Functions

- [Categories](#) ()
The default Constructor.
- [Categories](#) (List< [Category](#) > cats)
The constructor to use when a list of categories is given.
- bool [AddCategory](#) ([Category](#) c)
Method used to add a category to a list of categories.
- bool [RemoveCategory](#) ([Category](#) c)
Method used to remove a category from a list of categories.
- bool [ExistCategory](#) (int id)
Method used to verify if a category exists on a list of makes, given its ID.
- bool [ExistCategory](#) (string name)
Method used to verify if a Category exists on a list of categories, given its name.

Properties

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

5.4.1 Detailed Description

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

Definition at line 21 of file [Categories.cs](#).

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Categories() [1/2]

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

The default Constructor.

Definition at line 34 of file [Categories.cs](#).

5.4.2.2 Categories() [2/2]

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

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

Parameters

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

Definition at line 43 of file [Categories.cs](#).

5.4.3 Member Function Documentation

5.4.3.1 AddCategory()

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

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

Parameters

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

Returns

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

5.4.3.2 ExistCategory() [1/2]

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

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

Parameters

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

Returns

Definition at line 103 of file [Categories.cs](#).

5.4.3.3 ExistCategory() [2/2]

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

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

Parameters

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

Returns

Definition at line 120 of file [Categories.cs](#).

5.4.3.4 RemoveCategory()

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

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

Parameters

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

Returns

Definition at line 85 of file [Categories.cs](#).

5.4.4 Property Documentation

5.4.4.1 Cats

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

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

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

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

- [Category/Categories.cs](#)

5.5 trabalhoPOO_27967.Category Class Reference

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

Public Member Functions

- [Category](#) ()
The default Constructor.
- [Category](#) (string name)
The constructor to use when the name of a category is given.
- override bool [Equals](#) (object obj)
Method that overrides Equals() and verifies if a 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.

5.5.1 Detailed Description

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

Definition at line 20 of file [Category.cs](#).

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Category() [1/2]

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

The default Constructor.

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

5.5.2.2 Category() [2/2]

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

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

Parameters

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

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

5.5.3 Member Function Documentation

5.5.3.1 Equals()

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

Method that overrides Equals() and verifies if a 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](#).

5.5.3.2 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](#).

5.5.3.3 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](#).

5.5.4 Property Documentation

5.5.4.1 Id

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

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

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

5.5.4.2 Name

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

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

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

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

- [Category/Category.cs](#)

5.6 trabalhoPOO_27967.Client Class Reference

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

Public Member Functions

- [Client](#) ()
The default Constructor.
- [Client](#) (string n, string c)
Constructor to use when a name and a contact are given.
- override string [ToString](#) ()
Redefine the ToString Function to show a client's info.
- override bool [Equals](#) (object obj)
Redefine the Equals operator to verify if a client matches the other.

Static Public Member Functions

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

Properties

- int [ClientID](#) [get, set]
Property that sets or returns the ID of a client.
- string [Name](#) [get, set]
Property that sets or returns the Name of a client.
- string [Contact](#) [get, set]
Property that sets or returns the Contact of a client.
- static int [ClientCount](#) [get, set]
Property that sets or returns the amount of clients.

5.6.1 Detailed Description

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

Definition at line 23 of file [Client.cs](#).

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Client() [1/2]

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

The default Constructor.

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

5.6.2.2 Client() [2/2]

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

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

Parameters

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

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

5.6.3 Member Function Documentation

5.6.3.1 Equals()

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

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

Parameters

<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](#).

5.6.3.2 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](#).

5.6.3.3 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](#).

5.6.3.4 ToString()

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

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

Returns

Definition at line 116 of file [Client.cs](#).

5.6.4 Property Documentation

5.6.4.1 ClientCount

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

Property that sets or returns the amount of clients.

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

5.6.4.2 ClientID

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

Property that sets or returns the ID of a client.

Definition at line 65 of file [Client.cs](#).

5.6.4.3 Contact

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

Property that sets or returns the Contact of a client.

Definition at line 83 of file [Client.cs](#).

5.6.4.4 Name

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

Property that sets or returns the Name of a client.

Definition at line 74 of file [Client.cs](#).

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

- [Client/Client.cs](#)

5.7 trabalhoPOO_27967.Clients Class Reference

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

Public Member Functions

- [Clients](#) ()
The default Constructor.
- bool [AddClient](#) ([Client](#) cli)
Method to add a client to the store's client list.
- bool [RemoveClient](#) ([Client](#) cli)
Method to remove a client from the store's client list.
- bool [ExistClient](#) (int id)
Method to check if a client is listed on a clients' list.
- [Client](#) [GetClient](#) (int id)
Method used to get a client from a clients' list.

Properties

- List< [Client](#) > [ClientList](#) [get, set]
Property used to get and set the client list.

5.7.1 Detailed Description

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

Definition at line 21 of file [Clients.cs](#).

5.7.2 Constructor & Destructor Documentation

5.7.2.1 Clients()

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

The default Constructor.

Definition at line 34 of file [Clients.cs](#).

5.7.3 Member Function Documentation

5.7.3.1 AddClient()

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

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

Parameters

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

Returns

Definition at line 65 of file [Clients.cs](#).

5.7.3.2 ExistClient()

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

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

Parameters

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

Returns

Definition at line 95 of file [Clients.cs](#).

5.7.3.3 GetClient()

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

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

Parameters

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

Returns

Definition at line 112 of file [Clients.cs](#).

5.7.3.4 RemoveClient()

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

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

Parameters

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

Returns

Definition at line 77 of file [Clients.cs](#).

5.7.4 Property Documentation

5.7.4.1 ClientList

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

Property used to get and set the client list.

Definition at line 47 of file [Clients.cs](#).

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

- Client/[Clients.cs](#)

5.8 trabalhoPOO_27967.Make Class Reference

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

Public Member Functions

- [Make](#) ()
The default Constructor.
- [Make](#) (string name)
Constructor when the name of the make is given.
- override string [ToString](#) ()
Override of the ToString() method to convert the data of a Make to a string.
- override bool [Equals](#) (object obj)
The redefinition of the Equals() Method, to verify if a make matches another one.

Static Public Member Functions

- static bool [operator==](#) ([Make](#) m1, [Make](#) m2)
The redefinition of the Equal operator.
- static bool [operator!=](#) ([Make](#) m1, [Make](#) m2)
The redefinition of the NOT Equal operator.

Properties

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

5.8.1 Detailed Description

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

Definition at line 23 of file [Make.cs](#).

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Make() [1/2]

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

The default Constructor.

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

5.8.2.2 Make() [2/2]

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

Constructor when the name of the make is given.

Parameters

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

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

5.8.3 Member Function Documentation

5.8.3.1 Equals()

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

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

Parameters

<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](#).

5.8.3.2 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](#).

5.8.3.3 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](#).

5.8.3.4 ToString()

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

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

Returns

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

5.8.4 Property Documentation

5.8.4.1 ID

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

Property to set and get the ID of a Make.

Definition at line 64 of file [Make.cs](#).

5.8.4.2 Name

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

Property to get and set the Name of a Make.

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

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

- [Make/Make.cs](#)

5.9 trabalhoPOO_27967.Makes Class Reference

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

Public Member Functions

- [Makes](#) ()
The default Constructor.
- [Makes](#) (List< [Make](#) > m)
The constructor to use when a list of Make is given.
- bool [AddMake](#) ([Make](#) m)
Method used to add a make to a list of makes.
- bool [RemoveMake](#) ([Make](#) m)
Method used to remove a make from a list of makes.
- bool [ExistMake](#) (int id)
Method used to verify if a make exists on a list of makes, given its ID.
- bool [ExistMake](#) (string name)
Method used to verify if a make exists on a list of makes, given its name.

Properties

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

5.9.1 Detailed Description

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

Definition at line 21 of file [Makes.cs](#).

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Makes() [1/2]

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

The default Constructor.

Definition at line 34 of file [Makes.cs](#).

5.9.2.2 Makes() [2/2]

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

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

Parameters

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

Definition at line 43 of file [Makes.cs](#).

5.9.3 Member Function Documentation

5.9.3.1 AddMake()

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

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

Parameters

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

Returns

Definition at line 71 of file [Makes.cs](#).

5.9.3.2 ExistMake() [1/2]

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

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

Parameters

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

Returns

Definition at line 101 of file [Makes.cs](#).

5.9.3.3 ExistMake() [2/2]

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

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

Parameters

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

Returns

Definition at line 118 of file [Makes.cs](#).

5.9.3.4 RemoveMake()

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

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

Parameters

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

Returns

Definition at line 83 of file [Makes.cs](#).

5.9.4 Property Documentation

5.9.4.1 MakeList

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

The property to get and set a list of Make.

Definition at line 53 of file [Makes.cs](#).

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

- [Make/Makes.cs](#)

5.10 trabalhoPOO_27967.Product Class Reference

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

Public Member Functions

- [Product](#) ()
The default Constructor.
- [Product](#) (string reff, decimal pri, [Warranty](#) war, [Make](#) ma, [Category](#) cat)
Constructor for when the reference, price and warranty duration are given.
- override bool [Equals](#) (object obj)
Redefinition of the method to compare two products.
- override string [ToString](#) ()
Override of the ToString() Method to convert the data of a product into a string.

Static Public Member Functions

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

Properties

- string `Reference` [get, set]
Property to set and get the reference of a product.
- decimal `Price` [get, set]
Property to get and set the price of a product.
- `Make Make` [get, set]
Property to set and get the Make of a product.
- `Category Category` [get, set]
Property to set and get the Category of a product.
- int `Stock` [get, set]
Property to get and set the existing stock of a product.
- `Warranty Warranty` [get, set]

5.10.1 Detailed Description

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

Definition at line 25 of file `Product.cs`.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `Product()` [1/2]

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

The default Constructor.

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

5.10.2.2 `Product()` [2/2]

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

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

Parameters

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

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

5.10.3 Member Function Documentation

5.10.3.1 Equals()

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

Redefinition of the method to compare two products.

Parameters

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

Returns

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

5.10.3.2 operator"!=()

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

Redefinition of the different operator.

Parameters

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

Returns

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

5.10.3.3 operator==()

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

Redefinition of the equal operator.

Parameters

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

Returns

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

5.10.3.4 ToString()

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

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

Returns

Definition at line 167 of file [Product.cs](#).

5.10.4 Property Documentation

5.10.4.1 Category

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

Property to set and get the Category of a product.

Definition at line 103 of file [Product.cs](#).

5.10.4.2 Make

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

Property to set and get the Make of a product.

Definition at line 93 of file [Product.cs](#).

5.10.4.3 Price

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

Property to get and set the price of a product.

Definition at line 84 of file [Product.cs](#).

5.10.4.4 Reference

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

Property to set and get the reference of a product.

Definition at line 74 of file [Product.cs](#).

5.10.4.5 Stock

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

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

Definition at line 112 of file [Product.cs](#).

5.10.4.6 Warranty

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

Definition at line 118 of file [Product.cs](#).

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

- Product/[Product.cs](#)

5.11 trabalhoPOO_27967.Products Class Reference

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

Public Member Functions

- [Products](#) ()
The default Constructor.
- [Products](#) (List< [Product](#) > products)
The constructor to use when list of Product is given.
- override string [ToString](#) ()
Override of the ToString() Method to convert the data of a list fo products to a string.
- decimal [ValueInPosition](#) (int p)
This method returns the price of a product, given a certain array position.
- bool [InsertProduct](#) ([Product](#) p)
This method inserts a product in a list of products.
- bool [ExistProduct](#) (string reff)
Method used to verify if a product is on a products' list.
- bool [RemoveProduct](#) ([Product](#) p)
Method used to remove a product from a Products' list.
- [Product SearchProduct](#) (string reff)
This method searches for a product in an array, given its 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.

5.11.1 Detailed Description

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

Definition at line 25 of file [Products.cs](#).

5.11.2 Constructor & Destructor Documentation

5.11.2.1 Products() [1/2]

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

The default Constructor.

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

5.11.2.2 Products() [2/2]

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

The constructor to use when list of Product is given.

Parameters

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

Definition at line 47 of file [Products.cs](#).

5.11.3 Member Function Documentation

5.11.3.1 ExistProduct()

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

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

Parameters

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

Returns

Definition at line 115 of file [Products.cs](#).

5.11.3.2 InsertProduct()

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

This method inserts a product in a list of products.

Parameters

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

Returns

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

Definition at line 103 of file [Products.cs](#).

5.11.3.3 RemoveProduct()

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

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

Parameters

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

Returns

Product removed successfully

Product was not removed.

Definition at line 129 of file [Products.cs](#).

5.11.3.4 SearchProduct()

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

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

Parameters

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

Returns

Returns the product if found

Definition at line 145 of file [Products.cs](#).

5.11.3.5 ToString()

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

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

Returns

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

5.11.3.6 TotalPrice()

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

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

Returns

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

5.11.3.7 ValueInPosition()

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

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

Parameters

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

Returns

Definition at line 93 of file [Products.cs](#).

5.11.3.8 WarratyExpirationDateForProduct()

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

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

Parameters

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

Returns

Definition at line 170 of file [Products.cs](#).

5.11.4 Property Documentation

5.11.4.1 Prods

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

Property used to get and set the list of products.

Definition at line 60 of file [Products.cs](#).

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

- [Product/Products.cs](#)

5.12 trabalhoPOO_27967.Sale Class Reference

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

Public Member Functions

- [Sale](#) ()
The default Constructor.
- [Sale](#) ([Client](#) client, [Products](#) products, [Campaign](#) camp)
Constructor used when a client, a product array and a campaign to be used are given.
- override bool [Equals](#) (object obj)
Redefinition of the Equals method to compare two sales.
- override string [ToString](#) ()
- decimal [TotalPrice](#) ()
Method to calculate the total price of a sale, given the products list and a campaign code.
- bool [InsertProductOnSale](#) ([Product](#) p)
Method used to insert a product on a sale's list.
- bool [RemoveProductFromSale](#) ([Product](#) p)
Method used to remove a product from a sale.
- bool [ExistProductOnSale](#) ([Product](#) p)
Method used to verify if a product is on a sale.
- DateTime [WarrantyExpirationDate](#) (string reff)
Method to calculate when a warranty is due to expire.

Static Public Member Functions

- static bool [operator==](#) ([Sale](#) s1, [Sale](#) s2)
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.

5.12.1 Detailed Description

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

Definition at line 25 of file [Sale.cs](#).

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Sale() [1/2]

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

The default Constructor.

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

5.12.2.2 Sale() [2/2]

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

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

Parameters

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

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

5.12.3 Member Function Documentation

5.12.3.1 Equals()

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

Redefinition of the Equals method to compare two sales.

Parameters

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

Returns

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

5.12.3.2 ExistProductOnSale()

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

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

Parameters

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

Returns

Definition at line 220 of file [Sale.cs](#).

5.12.3.3 InsertProductOnSale()

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

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

Parameters

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

Returns

Definition at line 200 of file [Sale.cs](#).

5.12.3.4 operator"!=()"

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

Redefinition of the different operator.

Parameters

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

Returns

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

5.12.3.5 operator==()

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

Redefiniiton of the equal operator.

Parameters

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

Returns

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

5.12.3.6 RemoveProductFromSale()

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

Method used to remove a product from a sale.

Parameters

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

Returns

Definition at line 210 of file [Sale.cs](#).

5.12.3.7 ToString()

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

\u20AC é o unicode para o símbolo de euro.

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

5.12.3.8 TotalPrice()

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

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

Parameters

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

Returns

Definition at line 181 of file [Sale.cs](#).

5.12.3.9 WarrantyExpirationDate()

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

Method to calculate when a warranty is due to expire.

Parameters

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

Returns

Definition at line 231 of file [Sale.cs](#).

5.12.4 Property Documentation

5.12.4.1 Campaigns

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

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

Definition at line 120 of file [Sale.cs](#).

5.12.4.2 Client

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

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

Definition at line 84 of file [Sale.cs](#).

5.12.4.3 Id

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

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

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

5.12.4.4 Products

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

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

Definition at line 93 of file [Sale.cs](#).

5.12.4.5 SaleDate

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

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

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

5.12.4.6 TotPrice

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

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

Definition at line 102 of file [Sale.cs](#).

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

- Sale/[Sale.cs](#)

5.13 trabalhoPOO_27967.Sales Class Reference

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

Public Member Functions

- [Sales](#) ()
The default Constructor.
- [Sales](#) (List< [Sale](#) > sales)
The constructor to use when the sales' List is given.
- [Sale](#) [GetSale](#) (int idSale)
Method to find a sale in a list of sales, given its ID.
- bool [AddSale](#) ([Sale](#) sale)
Method used to add a sale to a sales' list.

Properties

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

5.13.1 Detailed Description

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

Definition at line 21 of file [Sales.cs](#).

5.13.2 Constructor & Destructor Documentation

5.13.2.1 [Sales](#)() [1/2]

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

The default Constructor.

Definition at line 34 of file [Sales.cs](#).

5.13.2.2 [Sales](#)() [2/2]

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

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

Parameters

sales	
-----------------------	--

Definition at line 43 of file [Sales.cs](#).

5.13.3 Member Function Documentation

5.13.3.1 AddSale()

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

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

Parameters

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

Returns

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

5.13.3.2 GetSale()

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

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

Parameters

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

Returns

Definition at line 74 of file [Sales.cs](#).

5.13.4 Property Documentation

5.13.4.1 SalesStored

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

Property used to get and set a list of sales.

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

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

- [Sale/Sales.cs](#)

5.14 trabalhoPOO_27967.Store.Store Class Reference

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

Public Member Functions

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

Properties

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

5.14.1 Detailed Description

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

Definition at line 20 of file [Store.cs](#).

5.14.2 Constructor & Destructor Documentation

5.14.2.1 [Store\(\)](#) [1/2]

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

The default Constructor.

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

5.14.2.2 [Store\(\)](#) [2/2]

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

The constructor to use when all the lists are given.

Parameters

<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](#).

5.14.3 Property Documentation

5.14.3.1 CatList

[Categories](#) trabalhoPOO_27967.Store.Store.CatList [get], [set]

The property to get and set de categories' list.

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

5.14.3.2 ClientList

[Clients](#) trabalhoPOO_27967.Store.Store.ClientList [get], [set]

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

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

5.14.3.3 MakeList

[Makes](#) trabalhoPOO_27967.Store.Store.MakeList [get], [set]

The property to get and set de makes' list.

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

5.14.3.4 ProdList

[Products](#) trabalhoPOO_27967.Store.Store.ProdList [get], [set]

The property to get and set de products' list.

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

5.14.3.5 SaleList

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

The property to get and set de sales' list.

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

5.14.3.6 WarrantList

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

The property to get and set de warranties' list.

Definition at line 118 of file [Store.cs](#).

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

- Store/[Store.cs](#)

5.15 trabalhoPOO_27967.Warranties Class Reference

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

Public Member Functions

- [Warranties](#) ()
The default Constructor.
- [Warranties](#) (List< [Warranty](#) > warrants)
The constructor to use when a list of Warranties is given.
- bool [AddWarranty](#) ([Warranty](#) w)
- bool [RemoveWarranty](#) ([Warranty](#) w)
Method used to remove a warranty from a list of warranties.
- bool [ExistWarranty](#) (string id)
Method used to confirm if a campaign exists on a list of campaigns.

Properties

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

5.15.1 Detailed Description

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

Definition at line 21 of file [Warranties.cs](#).

5.15.2 Constructor & Destructor Documentation

5.15.2.1 Warranties() [1/2]

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

The default Constructor.

Definition at line 34 of file [Warranties.cs](#).

5.15.2.2 Warranties() [2/2]

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

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

Parameters

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

Definition at line 43 of file [Warranties.cs](#).

5.15.3 Member Function Documentation

5.15.3.1 AddWarranty()

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

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

Parameters

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

Returns

Definition at line 74 of file [Warranties.cs](#).

5.15.3.2 ExistWarranty()

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

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

Parameters

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

Returns

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

5.15.3.3 RemoveWarranty()

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

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

Parameters

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

Returns

Definition at line 86 of file [Warranties.cs](#).

5.15.4 Property Documentation**5.15.4.1 Warrants**

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

Property used to get and set the list of warranties.

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

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

- Warranty/[Warranties.cs](#)

5.16 trabalhoPOO_27967.Warranty Class Reference

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

Public Member Functions

- [Warranty](#) ()
The default Constructor.
- [Warranty](#) (string prodID, int durationInYears, string conditions)
The constructor to use when the ID of the product, duration (in years) and the conditions are given.
- override string [ToString](#) ()
Method to show the information of a Warranty.
- override bool [Equals](#) (object obj)
Redefine the Equals operator to verify if a warranty matches the other.
- DateTime [ExpirationDate](#) ([Sale](#) s, string reff)
Method to calculate when a warranty is due to expire.

Static Public Member Functions

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

Properties

- string [ProdID](#) [get, set]
Property to get and set the ID of the product to which the warranty is about.
- int [DurationInYears](#) [get, set]
The property to set and get the duration of a warranty (in years).
- string [Conditions](#) [get, set]
The property to get and set the terms of a warranty.

5.16.1 Detailed Description

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

Definition at line 23 of file [Warranty.cs](#).

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Warranty() [1/2]

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

The default Constructor.

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

5.16.2.2 Warranty() [2/2]

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

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

Parameters

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

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

5.16.3 Member Function Documentation

5.16.3.1 Equals()

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

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

Parameters

<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](#).

5.16.3.2 ExpirationDate()

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

Method to calculate when a warranty is due to expire.

Parameters

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

Returns

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

5.16.3.3 operator!=()

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

Redefinition of the != operator.

Parameters

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

Returns

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

5.16.3.4 operator==()

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

Redefinition of the == operator.

Parameters

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

Returns

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

5.16.3.5 ToString()

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

Method to show the information of a Warranty.

Returns

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

5.16.4 Property Documentation

5.16.4.1 Conditions

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

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

Definition at line 85 of file [Warranty.cs](#).

5.16.4.2 DurationInYears

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

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

Definition at line 76 of file [Warranty.cs](#).

5.16.4.3 ProdID

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

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

Definition at line 67 of file [Warranty.cs](#).

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

- Warranty/[Warranty.cs](#)

Chapter 6

File Documentation

6.1 BestSale.cs File Reference

Classes

- class [trabalhoPOO_27967.BestSale](#)

Namespaces

- namespace [trabalhoPOO_27967](#)

6.2 BestSale.cs

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

```
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 }

```

6.3 Campaign/Campaign.cs File Reference

Classes

- class [trabalhoPOO_27967.Campaign](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.4 Campaign.cs

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

```

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
00072     public int Id{
00073         get { return _id; }
00074         set { _id = value; }
00075     }
00076
00077     public string Name
00078     {
00079         get { return _name; }
00080         set { _name = value; }
00081     }
00082
00083     public decimal Discount
00084     {
00085         get { return _discount; }
00086         set { _discount = value; }
00087     }
00088
00089     public DateTime StartDate
00090     {
00091         get { return _startDate; }
00092         set { _startDate = value; }
00093     }
00094
00095     public DateTime EndDate
00096     {
00097         get { return _endDate; }
00098         set { _endDate = value; }
00099     }
00100
00101     public int CampaignCount
00102     {
00103         get { return _campaignCount; }
00104         set { _campaignCount = value; }
00105     }
00106
00107     #endregion
00108
00109     #region Overrides
00110     public override bool Equals(object obj)
00111     {
00112         if (obj == null)
00113         {
00114             return false;
00115         }
00116
00117         Campaign camp = obj as Campaign;
00118         return (this.Id == camp.Id && _name == camp.Name);
00119     }
00120
00121     public static bool operator ==(Campaign camp1, Campaign camp2)
00122     {
00123         return (camp1.Equals(camp2));
00124     }
00125
00126     public static bool operator !=(Campaign camp1, Campaign camp2)
00127     {
00128         return !(camp1.Equals(camp2));
00129     }
00130
00131     #endregion
00132
00133     #region OtherMethods
00134
00135     static public bool VerifyApplicability(Campaign camp)
00136     {
00137         if (camp.StartDate <= DateTime.Now && camp.EndDate >= DateTime.Now)
00138         {
00139             return true;
00140         }
00141
00142         return false;
00143     }
00144
00145     #endregion

```

```

00188
00189     #region Destructor
00193     ~Campaign()
00194     {
00195     }
00196     #endregion
00197
00198     #endregion
00199 }
00200 }

```

6.5 Campaign/Campaigns.cs File Reference

Classes

- class [trabalhoPOO_27967.Campaigns](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.6 Campaigns.cs

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

```

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
00013 namespace trabalhoPOO_27967
00014 {
00022     public class Campaigns
00023     {
00024         #region Attributes
00025         List<Campaign> _camps;
00026         #endregion
00027
00028         #region Methods
00029
00030         #region Constructors
00031
00035         public Campaigns()
00036         {
00037             _camps = new List<Campaign>();
00038         }
00039
00044         public Campaigns(List<Campaign> p) {
00045             _camps = p;
00046         }
00047
00048         #endregion
00049
00050         #region Properties
00054         public List<Campaign> Camps
00055         {
00056             get{ return _camps; }
00057             set{ _camps = value; }
00058         }
00059         #endregion
00060

```

```

00061
00062
00063     #region Overrides
00064
00065     #endregion
00066
00067     #region OtherMethods
00073     public bool AddCampaign(Campaign c)
00074     {
00075         if(c==null)return false;
00076         _camps.Add(c);
00077         return true;
00078     }
00079
00085     public bool RemoveCampaign(Campaign camp)
00086     {
00087         foreach (Campaign c in _camps)
00088         {
00089             if (c == camp)
00090             {
00091                 _camps.Remove(camp);
00092                 return true;
00093             }
00094         }
00095         return false;
00096     }
00097
00103     public bool ExistCampaign(int id)
00104     {
00105         foreach (Campaign c in _camps)
00106         {
00107             if (c.Id == id)
00108             {
00109                 return true;
00110             }
00111         }
00112         return false;
00113     }
00114     #endregion
00115
00116     #region Destructor
00120     ~Campaigns()
00121     {
00122     }
00123     #endregion
00124
00125     #endregion
00126 }
00127 }

```

6.7 Category/Categories.cs File Reference

Classes

- class [trabalhoPOO_27967.Categories](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.8 Categories.cs

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

```

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
00012 namespace trabalhoPOO_27967
00013 {
00021     public class Categories
00022     {
00023         #region Attributes
00024         List<Category> _cats;
00025         #endregion
00026
00027         #region Methods
00028
00029         #region Constructors
00030
00034         public Categories()
00035         {
00036             _cats = new List<Category>();
00037         }
00038
00043         public Categories(List<Category> cats)
00044         {
00045             _cats = cats;
00046         }
00047
00048
00049         #endregion
00050
00051         #region Properties
00055         public List<Category> Cats
00056         {
00057             get { return _cats; }
00058             set { _cats = value; }
00059         }
00060         #endregion
00061
00062
00063
00064         #region Overrides
00065         #endregion
00066
00067         #region OtherMethods
00073         public bool AddCategory(Category c)
00074         {
00075             if (c == null) return false;
00076             _cats.Add(c);
00077             return true;
00078         }
00079
00085         public bool RemoveCategory(Category c)
00086         {
00087             foreach (Category cate in _cats)
00088             {
00089                 if (cate == c)
00090                 {
00091                     _cats.Remove(cate);
00092                     return true;
00093                 }
00094             }
00095             return false;
00096         }
00097
00103         public bool ExistCategory(int id)
00104         {
00105             foreach (Category cate in _cats)
00106             {
00107                 if (cate.Id == id)
00108                 {
00109                     return true;
00110                 }
00111             }
00112             return false;
00113         }
00114
00120         public bool ExistCategory(string name)
00121         {
00122             foreach (Category cate in _cats)
00123             {
00124                 if (cate.Name == name)
00125                 {
00126                     return true;
00127                 }
00128             }
00129         }
00130     }
00131 }
```

```

00129         return false;
00130     }
00131
00132     #endregion
00133
00134     #region Destructor
00138     ~Categories()
00139     {
00140     }
00141     #endregion
00142
00143     #endregion
00144 }
00145 }

```

6.9 Category/Category.cs File Reference

Classes

- class [trabalhoPOO_27967.Category](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.10 Category.cs

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

```

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     {
00084         if (obj == null)
00085         {
00086             return false;
00087         }
00088
00090         Category cat = obj as Category;
00091         return (this.Id == cat.Id || _name == cat.Name);
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 }

```

6.11 Client/Client.cs File Reference

Classes

- class [trabalhoPOO_27967.Client](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.12 Client.cs

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

```

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
OU FORA?
00057         }
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
00100         public static int ClientCount
00101         {
00102             get { return _clientCount; }
00103
00104             set { _clientCount = value; }
00105         }
00106         #endregion
00107
00108         #region Overrides
00111
00116         public override string ToString()
00117         {
00118             StringBuilder sb = new StringBuilder();

```

```

00119         sb.AppendLine($"Client ID: {_clientID}");
00120         sb.AppendLine($"Name: {_name}");
00121         sb.AppendLine($"Contact: {_contact}");
00122
00123         return sb.ToString();
00124     }
00125
00131     public override bool Equals(object obj)
00132     {
00133         if (obj == null)
00134         {
00135             return false;
00136         }
00137
00138         Client client = obj as Client;
00140         return (this._clientID == client._clientID && _name == client._name);
00142     }
00143
00150     public static bool operator == (Client cli1, Client cli2)
00151     {
00152         return( cli1.Equals(cli2) );
00153     }
00154
00161     public static bool operator !=(Client cli1, Client cli2)
00162     {
00163         return !(cli1.Equals(cli2));
00164     }
00165     #endregion
00166
00167     #region OtherMethods
00169     #endregion
00170
00171     #region Destructor
00175     ~Client ()
00176     {
00177     }
00178     #endregion
00179
00180     #endregion
00181 }
00182 }

```

6.13 Client/Clients.cs File Reference

Classes

- class [trabalhoPOO_27967.Clients](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.14 Clients.cs

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

```

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

```



```
00012 namespace trabalhoPOO_27967
00013 {
00021     public class Clients
00022     {
00023         #region Attributes
00024         static List<Client> _clientList;
00025         #endregion
00026
00027         #region Methods
00028
00029         #region Constructors
00030
00034         public Clients()
00035         {
00036             _clientList = new List<Client>();
00037         }
00038
00039
00040         #endregion
00041
00042         #region Properties
00043
00047         public List<Client> ClientList
00048         {
00049             get{ return _clientList; }
00050             set{ _clientList = value; }
00051         }
00052         #endregion
00053
00054
00055
00056         #region Overrides
00057         #endregion
00058
00059         #region OtherMethods
00065         public bool AddClient(Client cli)
00066         {
00067             if (cli == null) return false;
00068             this.ClientList.Add(cli);
00069             return true;
00070         }
00071
00077         public bool RemoveClient(Client cli)
00078         {
00079             foreach(Client client in _clientList)
00080             {
00081                 if (client == cli)
00082                 {
00083                     _clientList.Remove(cli);
00084                     return true;
00085                 }
00086             }
00087             return false;
00088         }
00089
00095         public bool ExistClient(int id)
00096         {
00097             foreach (Client client in _clientList)
00098             {
00099                 if (client.ClientID == id)
00100                 {
00101                     return true;
00102                 }
00103             }
00104             return false;
00105         }
00106
00112         public Client GetClient(int id)
00113         {
00114             foreach (Client client in _clientList)
00115             {
00116                 if (client.ClientID == id)
00117                 {
00118                     return client;
00119                 }
00120             }
00121             return null;
00122         }
00123         #endregion
00124
00125         #region Destructor
00129         ~Clients()
00130         {
00131         }
00132         #endregion
00133
00134         #endregion
00134     }
```

```
00135     }
00136 }
```

6.15 Make/Make.cs File Reference

Classes

- class [trabalhoPOO_27967.Make](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.16 Make.cs

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

```
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     public override bool Equals(object obj)
00094     {
00101         if (obj == null)
00102         {
00103             return false;
00104         }
00105
00106         Make make = obj as Make;
00107         return (this.ID == make.ID || this.Name == make.Name);
00108     }
00109
00110     public static bool operator ==(Make m1, Make m2)
00111     {
00112         return (m1.Equals(m2));
00113     }
00114
00115     public static bool operator !=(Make m1, Make m2)
00116     {
00117         return !(m1.Equals(m2));
00118     }
00119
00120     #endregion
00121
00122     #region OtherMethods
00123     #endregion
00124
00125     #region Destructor
00141     ~Make()
00142     {
00143     }
00144     #endregion
00145
00146     #endregion
00147 }
00148 }

```

6.17 Make/Makes.cs File Reference

Classes

- class [trabalhoPOO_27967.Makes](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.18 Makes.cs

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

```

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
00012 namespace trabalhoP00_27967
00013 {
00021     public class Makes
00022     {
00023         #region Attributes
00024         List<Make> _makeList;
00025         #endregion
00026
00027         #region Methods
00028
00029         #region Constructors
00030
00034         public Makes()
00035         {
00036             _makeList = new List<Make>();
00037         }
00038
00043         public Makes(List<Make> m)
00044         {
00045             _makeList = m;
00046         }
00047         #endregion
00048
00049         #region Properties
00053         public List<Make> MakeList
00054         {
00055             get { return _makeList; }
00056             set { _makeList = value; }
00057         }
00058         #endregion
00059
00060
00061
00062         #region Overrides
00063         #endregion
00064
00065         #region OtherMethods
00071         public bool AddMake(Make m)
00072         {
00073             if (m == null) return false;
00074             _makeList.Add(m);
00075             return true;
00076         }
00077
00083         public bool RemoveMake(Make m)
00084         {
00085             foreach (Make make in _makeList)
00086             {
00087                 if (make == m)
00088                 {
00089                     _makeList.Remove(make);
00090                     return true;
00091                 }
00092             }
00093             return false;
00094         }
00095
00101         public bool ExistMake(int id)
00102         {
00103             foreach (Make make in _makeList)
00104             {
00105                 if (make.ID == id)
00106                 {
00107                     return true;
00108                 }
00109             }
00110             return false;
00111         }
00112
00118         public bool ExistMake(string name)
00119         {
00120             foreach (Make make in _makeList)
00121             {
00122                 if (make.Name == name)
00123                 {
00124                     return true;
00125                 }
00126             }
00127             return false;

```

```

00128     }
00129     #endregion
00130
00131     #region Destructor
00132     ~Makes()
00133     {
00134     }
00135     #endregion
00136
00137     #endregion
00138
00139     #endregion
00140
00141     }
00142 }

```

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

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

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

```

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")]

```

6.21 Product/Product.cs File Reference

Classes

- class [trabalhoPOO_27967.Product](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.22 Product.cs

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

```

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     #endregion
00067
00068     #region Properties
00069
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     #endregion
00124
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
00152     public static bool operator !=(Product p1, Product p2)
00153     {
00154         return !(p1.Equals(p2));
00155     }
00156
00157     public override string ToString()
00158     {
00159         StringBuilder sb = new StringBuilder();
00160         sb.AppendLine($"Product ID: {_reference}");
00161         sb.AppendLine($"Price: {_price}€");
00162         sb.AppendLine(_make.ToString());
00163         sb.AppendLine(_warranty.ToString());
00164
00165         return sb.ToString();
00166     }
00167
00168     #endregion
00169
00170     #region OtherMethods
00171     #endregion
00172
00173     #region Destructor
00174     ~Product()
00175     {
00176     }
00177     #endregion
00178
00179     #endregion
00180 }
00181
00182 }
```

6.23 Product/Products.cs File Reference

Classes

- class [trabalhoPOO_27967.Products](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.24 Products.cs

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

```

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 static System.Net.Mime.MediaTypeNames;
00015
00016  namespace trabalhoPOO_27967
00017  {
00025      public class Products
00026      {
```

```

00027         #region Attributes
00028         List<Product> _prods; //como faço uma propriedade para isto? Não consigo dar return do array,
certo?
00029         #endregion
00030
00031         #region Methods
00032
00033         #region Constructors
00034
00038         public Products()
00039         {
00040             _prods = new List<Product>();
00041         }
00042
00047         public Products(List<Product> products)
00048         {
00049             _prods = products;
00050         }
00051
00052
00053
00054         #endregion
00055
00056         #region Properties
00060         public List<Product> Prods
00061         {
00062             get { return _prods; }
00063             set { _prods = value; }
00064         }
00065         #endregion
00066
00067
00068
00069         #region Overrides
00070
00075         public override string ToString()
00076         {
00077             StringBuilder sb = new StringBuilder();
00078             foreach (var product in _prods)
00079             {
00080                 sb.AppendLine(product.ToString());
00081             }
00082             return sb.ToString();
00083         }
00084         #endregion
00085
00086         #region OtherMethods
00087
00093         public decimal ValueInPosition(int p)
00094         {
00095             return this.Prods[p].Price;
00096         }
00097
00103         public bool InsertProduct(Product p)
00104         {
00105             if (p == null) return false;
00106             _prods.Add(p);
00107             return true;
00108         }
00109
00115         public bool ExistProduct(string reff)
00116         {
00117             foreach (Product p in _prods)
00118             {
00119                 if (p.Reference == reff) return true;
00120             }
00121             return false;
00122         }
00123
00129         public bool RemoveProduct(Product p)
00130         {
00131             if (p == null) return false;
00132             if (ExistProduct(p.Reference))
00133             {
00134                 _prods.Remove(p);
00135                 return true;
00136             }
00137             return false;
00138         }
00139
00145         public Product SearchProduct(string reff)
00146         {
00147             foreach (Product p in _prods)
00148             {
00149                 if(p.Reference==reff) return p;
00150             }
00151             return null;

```



```

00152     }
00153
00158     public decimal TotalPrice()
00159     {
00160         return _prods.Sum(p => p.Price);
00161     }
00162
00163
00170     public DateTime WarratyExpirationDateForProduct(DateTime date, string reff)
00171     {
00172         Product p = SearchProduct(reff);
00173         {
00174             return (date.AddYears(p.Warranty.DurationInYears));
00175         }
00176     }
00177     #endregion
00178
00179     #region Destructor
00183     ~Products()
00184     {
00185     }
00186     #endregion
00187
00188     #endregion
00189 }
00190 }

```

6.25 Properties/AssemblyInfo.cs File Reference

6.26 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")]

```

6.27 Sale/Sale.cs File Reference

Classes

- class `trabalhoPOO_27967.Sale`

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.28 Sale.cs

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

```

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
00033          public Sale()
00034          {
00035              _products= new Products();
00036              _client= new Client();
00037              _campaigns= new Campaign();
00038          }
00039
00040          public Sale(Client client, Products products, Campaign camp)
00041          {
00042              _id = ++_numSales;
00043              _client = client;
00044              _products = products;
00045              _totPrice = TotalPrice();
00046              _saleDate = DateTime.Now;
00047          }
00048
00049          #endregion
00050
00051          #region Properties
00052          public int Id
00053          {
00054              get { return _id; }
00055              set { _id = value; }
00056          }
00057
00058          public Client Client
00059          {
00060              get { return (_client); }
00061              set { _client = value; }
00062          }
00063
00064          public Products Products
00065          {
00066              get { return _products; }
00067              set { _products = value; }
00068          }
00069
00070          public decimal TotPrice

```

```

00103     {
00104         get { return _totPrice; }
00105         set { _totPrice = value; }
00106     }
00107
00111     public DateTime SaleDate
00112     {
00113         get { return _saleDate; }
00114         set { _saleDate = value; }
00115     }
00116
00120     public Campaign Campaigns
00121     {
00122         get { return _campaigns; }
00123         set { _campaigns = value; }
00124     }
00125     #endregion
00126
00127
00128
00129     #region Overrides
00135     public override bool Equals(object obj)
00136     {
00137         Sale sale = obj as Sale;
00138         return (sale.Id==this.Id);
00139     }
00140
00147     public static bool operator ==(Sale s1, Sale s2)
00148     {
00149         return(s1.Equals(s2));
00150     }
00151
00158     public static bool operator !=(Sale s1, Sale s2)
00159     {
00160         return !(s1.Equals(s2));
00161     }
00162
00163     public override string ToString()
00164     {
00165         StringBuilder sb = new StringBuilder();
00166         sb.AppendLine(_products.ToString());
00167         sb.AppendLine("Total " + this.TotalPrice().ToString() + "\u20AC");
00168         return sb.ToString();
00169     }
00170
00171     #endregion
00172
00173     #region OtherMethods
00174
00181     public decimal TotalPrice()
00182     {
00183         decimal total = 0;
00184
00185         total=_products.TotalPrice();
00186
00187         if (Campaign.VerifyApplicability(this.Campaigns))
00188         {
00189             total *= (1-this.Campaigns.Discount);
00190         }
00191
00192         return total;
00193     }
00194
00200     public bool InsertProductOnSale(Product p)
00201     {
00202         return _products.InsertProduct(p);
00203     }
00204
00210     public bool RemoveProductFromSale(Product p)
00211     {
00212         return _products.RemoveProduct(p);
00213     }
00214
00220     public bool ExistProductOnSale(Product p)
00221     {
00222         return _products.ExistProduct(p.Reference);
00223     }
00224
00231     public DateTime WarrantyExpirationDate(string reff)
00232     {
00233         return (_products.WarrantyExpirationDateForProduct(this.SaleDate, reff));
00234     }
00235     #endregion
00236
00237     #region Destructor
00241     ~Sale()
00242     {

```

```

00243     }
00244     #endregion
00245
00246     #endregion
00247 }
00248 }

```

6.29 Sale/Sales.cs File Reference

Classes

- class [trabalhoPOO_27967.Sales](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.30 Sales.cs

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

```

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
00012 namespace trabalhoPOO_27967
00013 {
00021     public class Sales
00022     {
00023         #region Attributes
00024         static List<Sale> _salesStored;
00025         #endregion
00026
00027         #region Methods
00028
00029         #region Constructors
00030
00034         public Sales()
00035         {
00036             _salesStored = new List<Sale>();
00037         }
00038
00043         public Sales(List<Sale> sales)
00044         {
00045             _salesStored = sales;
00046         }
00047
00048
00049         #endregion
00050
00051         #region Properties
00055         public List<Sale> SalesStored
00056         {
00057             get { return _salesStored; }
00058             set { _salesStored = value; }
00059         }
00060
00061         #endregion
00062
00063
00064

```

```

00065         #region Overrides
00066         #endregion
00067
00068         #region OtherMethods
00074         public Sale GetSale(int idSale)
00075         {
00076             foreach (Sale s in _salesStored)
00077             {
00078                 if (s.Id == idSale) return s;
00079             }
00080             return null;
00081         }
00082
00088         public bool AddSale(Sale sale)
00089         {
00090             if (_salesStored == null) return false;
00091             _salesStored.Add(sale);
00092             return true;
00093         }
00094         #endregion
00095
00096         #region Destructor
00100         ~Sales()
00101         {
00102         }
00103         #endregion
00104
00105         #endregion
00106     }
00107 }

```

6.31 Store/Store.cs File Reference

Classes

- class [trabalhoPOO_27967.Store.Store](#)

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

Namespaces

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

6.32 Store.cs

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

```

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 }

```

6.33 Warranty/Warranties.cs File Reference

Classes

- class [trabalhoPOO_27967.Warranties](#)

Purpose: This file has the definition and methods to work with the plurality of Warranty. Created by: Jose Alves a27967

Created on: 11/14/2024 4:20:11 PM.

Namespaces

- namespace [trabalhoPOO_27967](#)

6.34 Warranties.cs

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

```

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
00012  namespace trabalhoPOO_27967
00013  {
00021      public class Warranties
00022      {
00023          #region Attributes
00024          List<Warranty> _warrants;
00025          #endregion
00026
00027          #region Methods
00028
00029          #region Constructors
00030
00034          public Warranties()
00035          {
00036              _warrants = new List<Warranty>();
00037          }
00038
00043          public Warranties(List<Warranty> warrants)
00044          {
00045              _warrants = warrants;
00046          }
00047
00048
00049
00050          #endregion
00051
00052          #region Properties
00056          public List<Warranty> Warrants
00057          {
00058              get { return _warrants; }
00059              set { _warrants = value; }
00060          }
00061          #endregion
00062
00063
00064
00065          #region Overrides
00066          #endregion
00067
00068          #region OtherMethods
00069          // <summary>
00074          public bool AddWarranty(Warranty w)
00075          {
00076              if (w == null) return false;
00077              _warrants.Add(w);
00078              return true;
00079          }

```

```

00080
00086     public bool RemoveWarranty(Warranty w)
00087     {
00088         foreach (Warranty war in _warrants)
00089         {
00090             if (war==w)
00091             {
00092                 _warrants.Remove(war);
00093                 return true;
00094             }
00095         }
00096         return false;
00097     }
00098
00104     public bool ExistWarranty(string id)
00105     {
00106         foreach (Warranty w in _warrants)
00107         {
00108             if (w.ProdID == id)
00109             {
00110                 return true;
00111             }
00112         }
00113         return false;
00114     }
00115     #endregion
00116
00117     #region Destructor
00121     ~Warranties()
00122     {
00123     }
00124     #endregion
00125
00126     #endregion
00127 }
00128 }

```

6.35 Warranty/Warranty.cs File Reference

Classes

- class [trabalhoPOO_27967.Warranty](#)

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

Namespaces

- namespace [trabalhoPOO_27967](#)

6.36 Warranty.cs

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

```

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
00034
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
00059
00060         #endregion
00061
00062         #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
00093
00094         #region Overrides
00099         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         }
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(ref f);
00160         return (s.SaleDate.AddYears(_durationInYears));
00161     }
00162     #endregion
00163
00164     #region Destructor
00165     ~Warranty()
00166     {
00167     }
00168     #endregion
00169
00170     #endregion
00171 }
00172
00173 }
```

Index

AddCampaign
 trabalhoPOO_27967.Campaigns, [16](#)
AddCategory
 trabalhoPOO_27967.Categories, [18](#)
AddClient
 trabalhoPOO_27967.Clients, [27](#)
AddMake
 trabalhoPOO_27967.Makes, [34](#)
AddSale
 trabalhoPOO_27967.Sales, [51](#)
AddWarranty
 trabalhoPOO_27967.Warranties, [55](#)

BestSale.cs, [61](#)

Campaign
 trabalhoPOO_27967.Campaign, [10](#)
Campaign/Campaign.cs, [62](#)
Campaign/Campaigns.cs, [64](#)
CampaignCount
 trabalhoPOO_27967.Campaign, [13](#)
Campaigns
 trabalhoPOO_27967.Campaigns, [15](#)
 trabalhoPOO_27967.Sale, [48](#)
Camps
 trabalhoPOO_27967.Campaigns, [17](#)
Categories
 trabalhoPOO_27967.Categories, [18](#)
Category
 trabalhoPOO_27967.Category, [21](#)
 trabalhoPOO_27967.Product, [38](#)
Category/Categories.cs, [65](#)
Category/Category.cs, [67](#)
CatList
 trabalhoPOO_27967.Store.Store, [53](#)
Cats
 trabalhoPOO_27967.Categories, [20](#)
Client
 trabalhoPOO_27967.Client, [24](#)
 trabalhoPOO_27967.Sale, [48](#)
Client/Client.cs, [68](#)
Client/Clients.cs, [70](#)
ClientCount
 trabalhoPOO_27967.Client, [26](#)
ClientID
 trabalhoPOO_27967.Client, [26](#)
ClientList
 trabalhoPOO_27967.Store.Store, [53](#)
ClientList
 trabalhoPOO_27967.Clients, [29](#)

Clients
 trabalhoPOO_27967.Clients, [27](#)
Conditions
 trabalhoPOO_27967.Warranty, [60](#)
Contact
 trabalhoPOO_27967.Client, [26](#)

Discount
 trabalhoPOO_27967.Campaign, [13](#)
DurationInYears
 trabalhoPOO_27967.Warranty, [60](#)

EndDate
 trabalhoPOO_27967.Campaign, [13](#)
Equals
 trabalhoPOO_27967.Campaign, [11](#)
 trabalhoPOO_27967.Category, [21](#)
 trabalhoPOO_27967.Client, [24](#)
 trabalhoPOO_27967.Make, [30](#)
 trabalhoPOO_27967.Product, [37](#)
 trabalhoPOO_27967.Sale, [45](#)
 trabalhoPOO_27967.Warranty, [58](#)
ExistCampaign
 trabalhoPOO_27967.Campaigns, [16](#)
ExistCategory
 trabalhoPOO_27967.Categories, [18](#), [19](#)
ExistClient
 trabalhoPOO_27967.Clients, [28](#)
ExistMake
 trabalhoPOO_27967.Makes, [34](#)
ExistProduct
 trabalhoPOO_27967.Products, [41](#)
ExistProductOnSale
 trabalhoPOO_27967.Sale, [46](#)
ExistWarranty
 trabalhoPOO_27967.Warranties, [55](#)
ExpirationDate
 trabalhoPOO_27967.Warranty, [58](#)

GetClient
 trabalhoPOO_27967.Clients, [28](#)
GetSale
 trabalhoPOO_27967.Sales, [51](#)

ID
 trabalhoPOO_27967.Make, [32](#)
Id
 trabalhoPOO_27967.Campaign, [14](#)
 trabalhoPOO_27967.Category, [22](#)
 trabalhoPOO_27967.Sale, [49](#)

- InsertProduct
 - trabalhoPOO_27967.Products, [41](#)
- InsertProductOnSale
 - trabalhoPOO_27967.Sale, [46](#)
- Make
 - trabalhoPOO_27967.Make, [30](#)
 - trabalhoPOO_27967.Product, [38](#)
- Make/Make.cs, [72](#)
- Make/Makes.cs, [73](#)
- MakeList
 - trabalhoPOO_27967.Makes, [35](#)
 - trabalhoPOO_27967.Store.Store, [53](#)
- Makes
 - trabalhoPOO_27967.Makes, [33](#)
- Name
 - trabalhoPOO_27967.Campaign, [14](#)
 - trabalhoPOO_27967.Category, [22](#)
 - trabalhoPOO_27967.Client, [26](#)
 - trabalhoPOO_27967.Make, [32](#)
- obj/Debug/.NETFramework,Version=v4.7.2.AssemblyAttributes.cs, [75](#)
- operator!=
 - trabalhoPOO_27967.Campaign, [11](#)
 - trabalhoPOO_27967.Category, [22](#)
 - trabalhoPOO_27967.Client, [25](#)
 - trabalhoPOO_27967.Make, [31](#)
 - trabalhoPOO_27967.Product, [37](#)
 - trabalhoPOO_27967.Sale, [46](#)
 - trabalhoPOO_27967.Warranty, [58](#)
- operator==
 - trabalhoPOO_27967.Campaign, [11](#)
 - trabalhoPOO_27967.Category, [22](#)
 - trabalhoPOO_27967.Client, [25](#)
 - trabalhoPOO_27967.Make, [31](#)
 - trabalhoPOO_27967.Product, [37](#)
 - trabalhoPOO_27967.Sale, [47](#)
 - trabalhoPOO_27967.Warranty, [59](#)
- Price
 - trabalhoPOO_27967.Product, [38](#)
- ProdID
 - trabalhoPOO_27967.Warranty, [60](#)
- ProdList
 - trabalhoPOO_27967.Store.Store, [53](#)
- Prods
 - trabalhoPOO_27967.Products, [43](#)
- Product
 - trabalhoPOO_27967.Product, [36](#)
- Product/Product.cs, [75](#)
- Product/Products.cs, [77](#)
- Products
 - trabalhoPOO_27967.Products, [40](#)
 - trabalhoPOO_27967.Sale, [49](#)
- Properties/AssemblyInfo.cs, [79](#)
- Reference
 - trabalhoPOO_27967.Product, [39](#)
- RemoveCampaign
 - trabalhoPOO_27967.Campaigns, [16](#)
- RemoveCategory
 - trabalhoPOO_27967.Categories, [19](#)
- RemoveClient
 - trabalhoPOO_27967.Clients, [28](#)
- RemoveMake
 - trabalhoPOO_27967.Makes, [35](#)
- RemoveProduct
 - trabalhoPOO_27967.Products, [41](#)
- RemoveProductFromSale
 - trabalhoPOO_27967.Sale, [47](#)
- RemoveWarranty
 - trabalhoPOO_27967.Warranties, [56](#)
- Sale
 - trabalhoPOO_27967.Sale, [45](#)
- Sale/Sale.cs, [79, 80](#)
- Sale/Sales.cs, [82](#)
- SaleDate
 - trabalhoPOO_27967.Sale, [49](#)
- SaleList
 - trabalhoPOO_27967.Store.Store, [53](#)
- Sales
 - trabalhoPOO_27967.Sales, [50](#)
- SalesStored
 - trabalhoPOO_27967.Sales, [51](#)
- SearchProduct
 - trabalhoPOO_27967.Products, [42](#)
- StartDate
 - trabalhoPOO_27967.Campaign, [14](#)
- Stock
 - trabalhoPOO_27967.Product, [39](#)
- Store
 - trabalhoPOO_27967.Store.Store, [52](#)
- Store/Store.cs, [83](#)
- ToString
 - trabalhoPOO_27967.Client, [25](#)
 - trabalhoPOO_27967.Make, [32](#)
 - trabalhoPOO_27967.Product, [38](#)
 - trabalhoPOO_27967.Products, [42](#)
 - trabalhoPOO_27967.Sale, [47](#)
 - trabalhoPOO_27967.Warranty, [59](#)
- TotalPrice
 - trabalhoPOO_27967.Products, [42](#)
 - trabalhoPOO_27967.Sale, [48](#)
- TotPrice
 - trabalhoPOO_27967.Sale, [49](#)
- trabalhoPOO_27967, [7](#)
- trabalhoPOO_27967.BestSale, [9](#)
- trabalhoPOO_27967.Campaign, [9](#)
 - Campaign, [10](#)
 - CampaignCount, [13](#)
 - Discount, [13](#)
 - EndDate, [13](#)
 - Equals, [11](#)
 - Id, [14](#)

- Name, [14](#)
- operator!=, [11](#)
- operator==, [11](#)
- StartDate, [14](#)
- VerifyApplicability, [13](#)
- trabalhoPOO_27967.Campaigns, [14](#)
 - AddCampaign, [16](#)
 - Campaigns, [15](#)
 - Camps, [17](#)
 - ExistCampaign, [16](#)
 - RemoveCampaign, [16](#)
- trabalhoPOO_27967.Categories, [17](#)
 - AddCategory, [18](#)
 - Categories, [18](#)
 - Cats, [20](#)
 - ExistCategory, [18, 19](#)
 - RemoveCategory, [19](#)
- trabalhoPOO_27967.Category, [20](#)
 - Category, [21](#)
 - Equals, [21](#)
 - Id, [22](#)
 - Name, [22](#)
 - operator!=, [22](#)
 - operator==, [22](#)
- trabalhoPOO_27967.Client, [23](#)
 - Client, [24](#)
 - ClientCount, [26](#)
 - ClientID, [26](#)
 - Contact, [26](#)
 - Equals, [24](#)
 - Name, [26](#)
 - operator!=, [25](#)
 - operator==, [25](#)
 - ToString, [25](#)
- trabalhoPOO_27967.Clients, [27](#)
 - AddClient, [27](#)
 - ClientList, [29](#)
 - Clients, [27](#)
 - ExistClient, [28](#)
 - GetClient, [28](#)
 - RemoveClient, [28](#)
- trabalhoPOO_27967.Make, [29](#)
 - Equals, [30](#)
 - ID, [32](#)
 - Make, [30](#)
 - Name, [32](#)
 - operator!=, [31](#)
 - operator==, [31](#)
 - ToString, [32](#)
- trabalhoPOO_27967.Makes, [32](#)
 - AddMake, [34](#)
 - ExistMake, [34](#)
 - MakeList, [35](#)
 - Makes, [33](#)
 - RemoveMake, [35](#)
- trabalhoPOO_27967.Product, [35](#)
 - Category, [38](#)
 - Equals, [37](#)
 - Make, [38](#)
 - operator!=, [37](#)
 - operator==, [37](#)
 - Price, [38](#)
 - Product, [36](#)
 - Reference, [39](#)
 - Stock, [39](#)
 - ToString, [38](#)
 - Warranty, [39](#)
- trabalhoPOO_27967.Products, [39](#)
 - ExistProduct, [41](#)
 - InsertProduct, [41](#)
 - Prods, [43](#)
 - Products, [40](#)
 - RemoveProduct, [41](#)
 - SearchProduct, [42](#)
 - ToString, [42](#)
 - TotalPrice, [42](#)
 - ValueInPosition, [42](#)
 - WarrantyExpirationDateForProduct, [43](#)
- trabalhoPOO_27967.Sale, [44](#)
 - Campaigns, [48](#)
 - Client, [48](#)
 - Equals, [45](#)
 - ExistProductOnSale, [46](#)
 - Id, [49](#)
 - InsertProductOnSale, [46](#)
 - operator!=, [46](#)
 - operator==, [47](#)
 - Products, [49](#)
 - RemoveProductFromSale, [47](#)
 - Sale, [45](#)
 - SaleDate, [49](#)
 - ToString, [47](#)
 - TotalPrice, [48](#)
 - TotPrice, [49](#)
 - WarrantyExpirationDate, [48](#)
- trabalhoPOO_27967.Sales, [49](#)
 - AddSale, [51](#)
 - GetSale, [51](#)
 - Sales, [50](#)
 - SalesStored, [51](#)
- trabalhoPOO_27967.Store, [8](#)
- trabalhoPOO_27967.Store.Store, [52](#)
 - CatList, [53](#)
 - ClientList, [53](#)
 - MakeList, [53](#)
 - ProdList, [53](#)
 - SaleList, [53](#)
 - Store, [52](#)
 - WarrantList, [54](#)
- trabalhoPOO_27967.Warranties, [54](#)
 - AddWarranty, [55](#)
 - ExistWarranty, [55](#)
 - RemoveWarranty, [56](#)
 - Warranties, [55](#)
 - Warrants, [56](#)
- trabalhoPOO_27967.Warranty, [56](#)

- Conditions, [60](#)
- DurationInYears, [60](#)
- Equals, [58](#)
- ExpirationDate, [58](#)
- operator!=, [58](#)
- operator==, [59](#)
- ProdID, [60](#)
- ToString, [59](#)
- Warranty, [57](#)

ValueInPosition

- trabalhoPOO_27967.Products, [42](#)

VerifyApplicability

- trabalhoPOO_27967.Campaign, [13](#)

Warranties

- trabalhoPOO_27967.Warranties, [55](#)

WarrantList

- trabalhoPOO_27967.Store.Store, [54](#)

Warrants

- trabalhoPOO_27967.Warranties, [56](#)

Warranty

- trabalhoPOO_27967.Product, [39](#)
- trabalhoPOO_27967.Warranty, [57](#)

Warranty/Warranties.cs, [85](#)

Warranty/Warranty.cs, [86](#)

WarrantyExpirationDate

- trabalhoPOO_27967.Sale, [48](#)

WarrantyExpirationDateForProduct

- trabalhoPOO_27967.Products, [43](#)