

Documentação sobre o trabalho dos pneus

1.1 Algoritmo Antigo

No semestre anterior (2º semestre) na disciplina de linguagem de programas e algoritmos foi feito um algoritmo onde na proposta de programa era pedido:

Numa inspeção veicular na frota de uma empresa, constatou que vários veículos teriam que efetuar a troca de pneus. Faça um fluxograma e um algoritmo que, leia a quantidade de veículos a serem inspecionados e seus respectivos códigos de cada veículo, calcule e mostre:

- A quantidade de pneus a serem substituídos por cada veículo.
- O valor do pneus para determinado veículo.
- O valor total de pneus para cada veículo.
- Quantos veículos terão que substituir os 4 pneus.
- Quantos veículos terão que substituir os 3 pneus.
- Quantos veículos terão que substituir os 2 pneus.
- Quantos veículos terão que substituir os 1 pneu.
- Quantos veículos terão que não precisarão substituir nenhum pneu.
- Qual o total geral de pneus a serem substituídos.
- Qual o total geral a ser gasto em toda essa operação de substituição dos pneus de todos os veículos.

Agora abaixo irei deixar o código que foi feito em português:

// Disciplina : [LPA]

// Professor : Julio Cesar

// Descrição : Aqui você descreve o que o programa faz! (função)

// Autor(a) : Nome do(a) aluno(a)

// Data atual : 19/10/2018

Var

// Seção de Declarações das variáveis

n, cont, qp, p1, p2, p3, p4, qpp : inteiro

vp, vt, vtt, vtt1, vtt2, vtt3, vtt4: real

Inicio

// Seção de Comandos, procedimento, funções, operadores, etc...

escreva("Quantidade de veículos: ")

leia(n)

cont := 1

p1 := 0

p2 := 0

p3 := 0

p4 := 0

vtt1 := 0

vtt2 := 0

vtt3 := 0

vtt4 := 0

enquanto cont <= n faça

 escreval("")

 escreva("Quantia de pneus a serem trocados: ")

 leia(qp)

 se(qp = 1) entao

 escreva("Qual o valor de cada pneu: ")

 leia(vp)

 vt <- vp * qp

 escreval("O valor da troca foi: ", vt)

```
p1 := p1 + 1  
vtt1 := vtt1 + vt  
fimse
```

```
se(qp = 2) entao  
    escreva("Qual o valor de cada pneu: ")  
    leia(vp)  
    vt <- vp * qp  
    escreval("O valor da troca foi: " ,vt)  
    p2 := p2 + 1  
    vtt2 := vtt2 + vt  
fimse
```

```
se(qp = 3) entao  
    escreva("Qual o valor de cada pneu: ")  
    leia(vp)  
    vt <- vp * qp  
    escreval("O valor da troca foi: " ,vt)  
    p3 := p3 + 1  
    vtt3 := vtt3 + vt  
fimse
```

```
se(qp = 4) entao  
    escreva("Qual o valor de cada pneu: ")  
    leia(vp)  
    vt <- vp * qp  
    escreval("O valor da troca foi: " ,vt)  
    p4 := p4 + 1  
    vtt4 := vtt4 + vt  
fimse
```

$qpp := p1 + (p2 * 2) + (p3 * 3) + (p4 * 4)$

$vtt := vtt1 + vtt2 + vtt3 + vtt4$

$cont := cont + 1$

fimenquanto

escreval("A quantidade total de pneus trocados foi de: ", qpp)

escreval("O valor total gasto foi de: ", vtt)

fimalgoritmo

1.2 O fluxograma

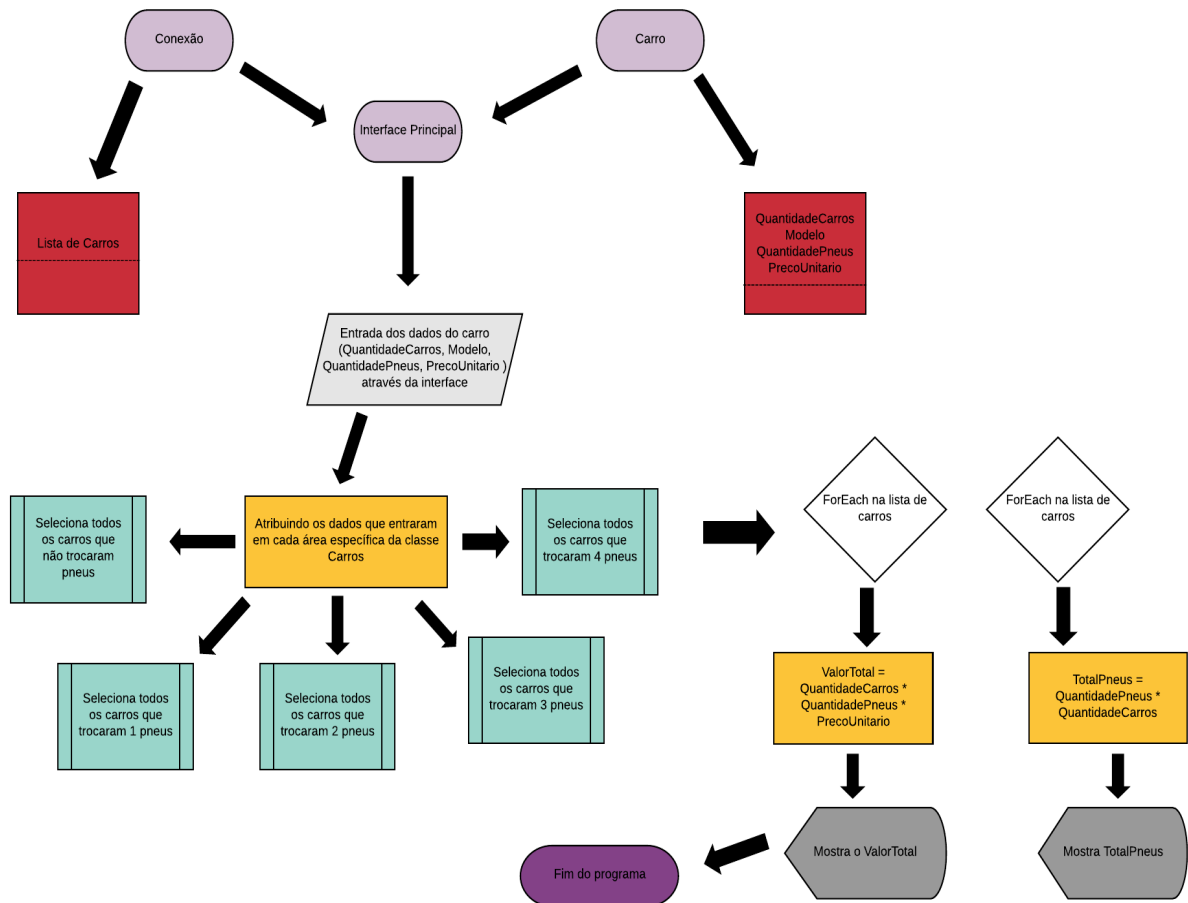
No antigo semestre como no atual foi pedido para que fosse feito um fluxograma do programa, porém não tenho acesso ao antigo fluxograma pois foi feito em uma folha em sala de aula.

Já nesse semestre foi feito um fluxograma digital informando os passos que foram seguidos para criação do programa.

Vale a pena ressaltar a importância do fluxograma para a criação de um programa, já que há muitos ganhos quando é utilizado da forma certa, entre elas:

- Facilita a organização do raciocínio;
- Mais representativo e fácil de ser lido do que processo escrito;
- Facilita a elaboração de atividades e tarefas;
- Controla os procedimentos;
- Pontos de verificação, decisão, entre outros;
- Localizar elementos que são desconexos ou perdidos

Aqui foi citado só algumas das vantagens da criação do fluxograma existe várias outras, abaixo está o fluxograma que foi criado para este programa específico:

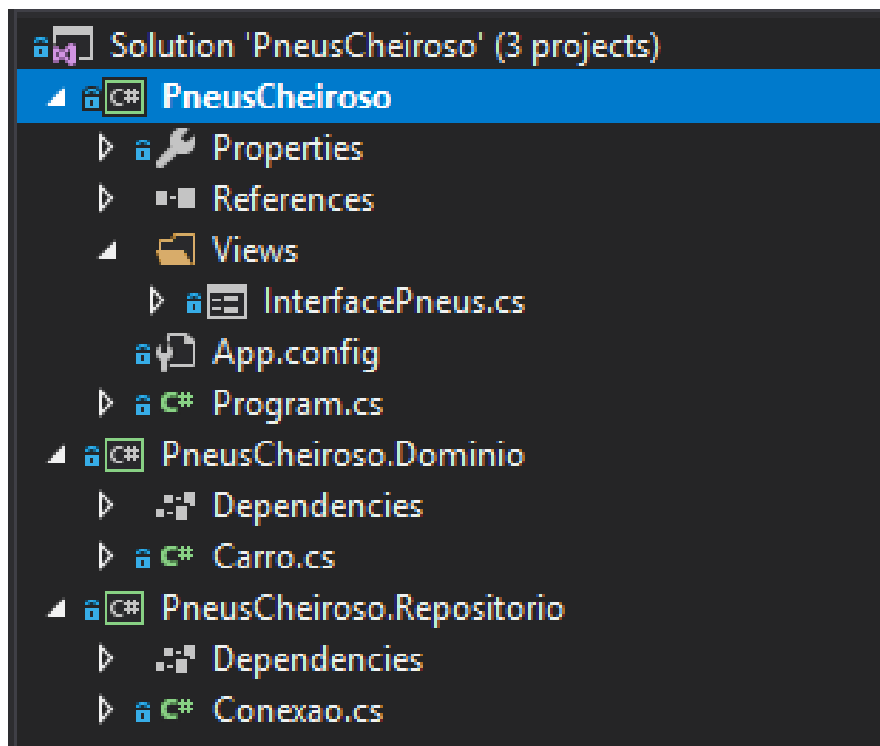


1.3 Criação do Programa

Para a criação do programa foi utilizada a linguagem de programação C# que é multiparadigma, de tipagem forte. Foi criada pela Microsoft como parte da plataforma .NET, a sintaxe principal que é orientada a objetos foi baseada no C++ mas inclui influências de outras linguagens como Java.

1.3.1 Divisão do Projeto

O trabalho foi dividido em três partes



1.3.2 PneusCheiroso

O projeto principal que é “PneusCheiroso” onde tem a main do programa, depois foi criado a pasta “Views” e dentro dela está “InterfacePneus” que é a tela que irá interagir com o usuário.



1.3.3 PneusCheiroso.Dominio

A partir disso foi criado um segundo projeto dentro da mesma solução, essa parte de domínio é onde se tem a Classe Carro, onde contém todos os atributos que são de um carro que aparecem na interface. Ou seja são atributos como Quantidade de Veículos, Modelo, Quantidade de Pneus e o Preço Unitário de cada pneu.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace PneusCheiroso.Dominio
6  {
7      6 references
7      public class Carro
8      {
9          3 references
9          public int QuantidadeCarros { get; set; }
10         1 reference
10         public string Modelo { get; set; }
11         8 references
11         public int QuantidadePneu { get; set; }
12         2 references
12         public double PrecoUnitario { get; set; }
13     }
14 }
15
```

1.3.4 PneusCheiroso.Repositorio

Como é dito no próprio nome é um repositório, onde tem a classe Conexao, e nessa classe tem-se uma lista onde é armazenada toda a informação sobre cada carro, ou seja, os atributos da classe Carro são diretamente colocados nesta lista e assim ficando salvo apenas na memória pois não há conexão com o banco de dados

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using PneusCheiroso.Dominio;
5
6  namespace PneusCheiroso.Repositorio
7  {
8      public class Conexao
9      {
10         public List<Carro> Carros = new List<Carro>();
11     }
12 }
13

```

1.3.5 Código

Abaixo vou colocar todo o código do programa

Interface

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using PneusCheiroso.Dominio;
using PneusCheiroso.Repositorio;

namespace PneusCheiroso
{
    public partial class InterfacePneus : Form
    {
        List<Carro> quantT = new List<Carro>();
        Carro dadosCarro;
        Conexao db = new Conexao();
        public InterfacePneus()
        {
            InitializeComponent();
        }

        private void Minimize_Click(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Minimized;
        }

        private void Close_Click(object sender, EventArgs e)
        {
            InterfacePneus.ActiveForm.Close();
        }
    }
}

```



```

private void Button1_Click(object sender, EventArgs e)
{
    dadosCarro = new Carro();
    dadosCarro.Modelo = textModelo.Text;
    dadosCarro.QuantidadeCarros = int.Parse(textQuantidade.Text);
    dadosCarro.QuantidadePneu = int.Parse(textQtdPneus.Text);
    dadosCarro.PrecoUnitario = double.Parse(textPreco.Text);

    db.Carros.Add(dadosCarro);
    quantT = db.Carros.ToList();
    GridCarros.DataSource = quantT;
}

private void Button5_Click(object sender, EventArgs e)
{
    quantT = db.Carros.Where(x => x.QuantidadePneu == 0).ToList();
    GridCarros.DataSource = quantT;
}

private void Button4_Click(object sender, EventArgs e)
{
    quantT = db.Carros.Where(x => x.QuantidadePneu == 1).ToList();
    GridCarros.DataSource = quantT;
}

private void Button3_Click(object sender, EventArgs e)
{
    quantT = db.Carros.Where(x => x.QuantidadePneu == 2).ToList();
    GridCarros.DataSource = quantT;
}

private void Button2_Click(object sender, EventArgs e)
{
    quantT = db.Carros.Where(x => x.QuantidadePneu == 3).ToList();
    GridCarros.DataSource = quantT;
}

private void Button6_Click(object sender, EventArgs e)
{
    quantT = db.Carros.Where(x => x.QuantidadePneu == 4).ToList();
    GridCarros.DataSource = quantT;
}

private void Button7_Click(object sender, EventArgs e)
{
    //lb_valorTotal.Text = "Valor Total: ";
    //lb_valorTotal.Text += "R$ " + ValorTotal().ToString();

    string message = "O preço total dos pneus foi: R$ " +
        ValorTotal().ToString(CultureInfo.InvariantCulture);
    string title = "Preço Total:";

    MessageBox.Show(message, title, MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}

private double ValorTotal()
{
    double total=0.0;

    foreach(var i in quantT)
    {
        total += i.QuantidadeCarros * i.QuantidadePneu * i.PrecoUnitario;
    }
}

```

```

        }
        return total;
    }

    private int TotalPneus()
    {
        int total = 0;

        foreach (var i in quantT)
        {
            total += i.QuantidadeCarros * i.QuantidadePneu;
        }

        return total;
    }

    private void Button8_Click(object sender, EventArgs e)
    {
        string message = "O total de pneus trocados em toda a frota foi de: "
+ TotalPneus().ToString(CultureInfo.InvariantCulture);
        string title = "Número total de pneus";

        MessageBox.Show(message, title, MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }

    private void Button9_Click(object sender, EventArgs e)
    {
        quantT = db.Carros.ToList();
        GridCarros.DataSource = quantT;
    }
}

```

Classe Carro

```

using System;
using System.Collections.Generic;
using System.Text;

namespace PneusCheiroso.Dominio
{
    public class Carro
    {
        public int QuantidadeCarros { get; set; }
        public string Modelo { get; set; }
        public int QuantidadePneu { get; set; }
        public double PrecoUnitario { get; set; }
    }
}

```

Classe Conexao

```

using System;
using System.Collections.Generic;
using System.Text;
using PneusCheiroso.Dominio;

```

```

namespace PneusCheiroso.Repositorio
{
    public class Conexao
    {
        public List<Carro> Carros = new List<Carro>();
    }
}

```

Arquivo Program (Main)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PneusCheiroso
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new InterfacePneus());
        }
    }
}

```