

POO I - Lista Exercícios 02

Prof Angel Antonio ¹

Resumo

Esta lista de exercícios consta de pequenos programas introdutórios. Objetivo principal é iniciar um contato com a linguagem C# e com a ferramenta de desenvolvimento Visual Studio. São apresentados também os conceitos iniciais sobre classes e objetos, que são elemento fundamentais do paradigma programação de orientada a objetos.

Palavras Chave

Programação Orientada ao Objeto — Linguagem C#

¹Instituto de Ciências Exatas e Tecnologia (ICET). Universidade Paulista (UNIP), São Paulo, SP, Brasil

Sumário

Lista de Exercícios 02	1
1.1 Parte I - Programação Estruturada	1
1.2 Parte II - Programação Orientada a Objetos	2
1.3 Parte III - Extra Windows Forms e Web Forms	4
Agradecimentos	4
Formatação	4
Referências	4

Lista de Exercícios 02

Esta lista de exercícios está dividida em três partes. A primeira parte é baseada na referência bibliográfica [1] que consiste de exercícios em C# na forma de programas no paradigma de programação estruturada. A segunda parte é baseada na referência [2] da mesma autora porém com programas no paradigma de programação orientada a objetos. Há uma parte três que não é objetivo principal deste curso porém serve como motivação ao estudante para a busca de outras formas de programação.

OBS: Texto não revisado

1.1 Parte I - Programação Estruturada

Exercício 1.

O programa deve formatar as duas frases de tal forma que as quantidades fiquem alinhadas em 10 espaços pela esquerda.

- a) Bananas 457 b) Peras 38

Exercício 2.

O programa deve formatar as duas frases de tal forma que as quantidades fiquem alinhadas em 10 espaços pela direita.

- a) Bananas 457 b) Peras 38

Exercício 3.

O programa deve executar as seguintes funções matemáticas.

- a) Seno de 30° b) Raiz quadrada de 4

Exercício 4.

O programa deve executar as seguintes operações lógicas.

- a) $4 == 5$ b) $4! = 6$
c) $4 > 5$ d) $4 < 5 \ \&\& \ 6 > 10$
e) $40 < 50 \ || \ 60 > 90$ f) $!(40 < 50 \ || \ 60 > 90)$

Exercício 5.

O programa deve executar as seguintes manipulações com strings.

- a) Determine o comprimento da string “**Ordem e Progresso**”.
- b) Concatene as string “**A capital do Brasil**” com “**é Brasília**”.
- c) Converta a string “no futuro teremos o homem terá uma base em marte” em letras maiúsculas.
- d) Extraia a sub string “**luz**” da frase “**A velocidade da luz não é o limite**”.
- e) Determine onde começa a sub string “**afundou**” na frase “**O Titanic afundou após colidir com um iceberg**”
- f) Compare em termos alfabéticos as palavras as palavras “**Piau**” e “**Goias**”
- g) Remova os espaços a no incio e no final da frase “**a lua alteras as marés**”

Exercício 6.

O programa deve atribuir e exibir o valor das variáveis apresentadas a seguir. Faça uma investigação do significado de inteiros sem sinal e com sinal.

- a) sbyte S=-34;
- b) byte B1=250;
- c) short SH=32000;
- d) ushort U=60000;
- e) int I=1000000012;
- f) uint UI=3500000000;
- g) long L=1532333333334581239;
- h) ulong UL=14532333333334581239;
- i) float F=345.3456f;
- j) double D=6.89765432127866;
- k) decimal DE=1234567897654300.14567896543m;
- l) char C='O';
- m) string ST="FCP é o Maior";
- n) bool LV=true;
- o) bool LF=false;

Exercício 7.

O programa deve executar as seguintes conversões de tipos de dados.

- a) Converta o double 99.56 para inteiro com um cast.
- b) Converta a string "99" para inteiro, e some 1 a este valor, então converta esse inteiro para string verificando a quantidade de dígitos do resultado.

Exercício 8.

Pesquise sobre as formatações e arredondamentos no itens a seguir.

- a) Exiba 3 formas de exibir um número de forma arredondada.
- b) Exibe 3 formas de exibir números em porcentagens.
- c) Exibe 5 formas de exibir a data e hora atuais do sistema.

Exercício 9.

Calcule o itens solicitados a seguir usando os operadores na forma reduzida.

- a) Se o salario de um funcionário é de mil unidades monetárias e recebeu dez unidades de aumento qual o salario atual.
- b) Se o valor de total das vendas foram de cem unidades monetárias e os impostos foram de vinte unidades qual o valor líquido.
- c) Se você investiu um capital de mil unidades monetárias e ao final do investimento recebeu cinco por cento qual o total do capital ao final do investimento.
- d) Se você possuía 10 unidades monetárias e as dividiu igualmente entre seus três colegas, ficando com o resto, com quanto você ficou.

Exercício 10.

Conhecendo o operador de incremento (++) e de decremento (--) faça um programa que teste todas as possibilidades de manipulação destes operadores.

1.2 Parte II - Programação Orientada a Objetos**Exercício 11.**

Construa um programa onde exista uma classe publica de nome **Funcionario**. A classe **Funcionario** deve ter um atributo publico do tipo inteiro e de nome **NumeroMatricula**. No programa crie um objeto da classe **Funcionario** de nome *funcionario*. Ao objeto *funcionario* atribua o valor 10 ao atributo **NumeroMatricula**, em seguida exiba o numero de matricula na tela do console.

Exercício 12.

Construa um programa onde exista uma classe publica de nome **Funcionario**. A classe **Funcionario** deve ter um atributo publico do tipo inteiro e de nome **NumeroMatricula**. Ainda na classe **Funcionario** crie um construtor com um argumento do tipo inteiro de nome *Matricula*, no interior do construtor inicialize o atributo **NumeroMatricula** com o valor do parâmetro *Matricula*. No programa crie um objeto de nome *funcionario* da classe **Funcionario** passando o valor 10 para o construtor, em seguida exiba o numero de matricula na tela do console.

Exercício 13.

Construa um programa onde exista uma classe publica de nome **Funcionario**. A classe **Funcionario** deve ter um atributo *privado* do tipo inteiro e de nome **NumeroMatricula**. Ainda na classe **Funcionario** crie um construtor com um argumento do tipo inteiro de nome *Matricula*, no interior do construtor inicialize o atributo

NumeroMatricula com o valor do parâmetro *Matricula*. Por fim na classe **Funcionario** crie um metodo de nome *getNumeroMatricula* que deve retornar o valor do atributo **NumeroMatricula**. No programa crie um objeto de nome *funcionario* da classe **Funcionario** passando o valor 10 para o construtor, em seguida exiba o numero de matricula na tela do console.

Exercício 14.

Construa um programa onde exista uma classe publica de nome **Funcionario**. A classe **Funcionario** deve ter um atributo *privado* do tipo inteiro e de nome **NumeroMatricula**. Ainda na classe **Funcionario** crie um construtor com um argumento do tipo inteiro de nome *Matricula*, no interior do construtor. Adicione um construtor sem parâmetros que inicialize a o atributo **NumeroMatricula** com o valor zero. inicialize o atributo **NumeroMatricula** com o valor do parâmetro *Matricula*. Por fim na classe **Funcionario** crie um metodo de nome *getNumeroMatricula* que deve retornar o valor do atributo **NumeroMatricula** e um método de nome *setNumeroMatricula* sem retorno e que possua um parâmetro do tipo inteiro de nome **Matricula**. O Método *setNumeroMatricula* deve atribuir ao atributo **NumeroMatricula** o valor do argumento **Matricula** recebido pelo método. No programa crie um objeto de nome *funcionario* da classe **Funcionario** usando o construtor sem argumentos, depois chame o método *setNumeroMatricula* passando o valor de 10 para o argumento **Matricula**, e em seguida exiba o numero de matricula na tela do console.

Exercício 15.

Construa um programa onde exista uma classe publica de nome **Funcionario**. A classe **Funcionario** deve ter um atributo privado do tipo inteiro e de nome **numeroMatricula**. Adicione à classe uma propriedade publicas de nome **NumeroMatricula** que seja de leitura e escrita em relação ao atributo **numeroMatricula**. Ainda na classe **Funcionario** crie um construtor com um argumento do tipo inteiro de nome *Matricula*, no interior do construtor inicialize o atributo **numeroMatricula** através da propriedade **NumeroMatricula** com o valor do parâmetro *Matricula*. Crie ainda um construtor sem parâmetros que inicialize a atributo **numeroMatricula** através da propriedade **NumeroMatricula** com o valor zero. No programa crie um objeto de nome *funcionario* da classe **Funcionario** passando o valor 10 para o construtor, em seguida exiba o numero de matricula na tela do console.

Exercício 16.

Construa um programa onde exista uma classe aluno. A classe aluno deve ter três atributos privados, um para o nome do aluno e outros dois para duas notas de prova. A classe deve ter um construtor que permita inicializar

estes três atributos. Adicione a classe aluno um método que calcule a média das notas e faça o arredondamento adequado da nota para o inteiro mais próximo (Pesquise para achar essa característica no C#). Exemplo: se o calculo da média der 5.3 a média será 5, porém de for 5.5 a média final será 6.

Exercício 17.

Construa um programa onde exista uma classe aluno. A classe aluno deve ter três atributos privados, um para o nome do aluno e outros dois para duas notas de prova. Para cada um destes atributos privados crie uma propriedade publica de apenas leitura. A classe deve ter um construtor que permita inicializar estes três atributos. Adicione a classe aluno um método que calcule a média das notas e faça o arredondamento adequado da nota para o inteiro mais próximo (Pesquise para achar essa característica no C#). Exemplo: se o calculo da média der 5.3 a média será 5, porém de for 5.5 a média final será 6.

Exercício 18.

Construa um programa onde exista uma classe aluno. A classe aluno deve ter três atributos privados, um para o nome do aluno e outros dois para duas notas de prova. Para cada um destes atributos privados crie uma propriedade publica de apenas leitura e escrita. A classe deve ter um construtor que permita inicializar estes três atributos e um construtor sem parâmetros que “limpe” as variáveis, ou seja, inicialize o nome com uma string vazia e as notas com zero. Adicione a classe aluno um método que calcule a média das notas e faça o arredondamento adequado da nota para o inteiro mais próximo (Pesquise para achar essa característica no C#). No programa use o construtor sem parâmetros para criar um objeto da classe aluno e inicialize os atributos através das propriedades Exemplo: se o calculo da média der 5.3 a média será 5, porém de for 5.5 a média final será 6.

Exercício 19.

Construa um programa onde exista uma classe **Area** com um método que calcule a área de um quadrado. No programa use a classe para calcular a área de um quadrado de lado 10.

Exercício 20.

Construa um programa para calcular a temperatura Fahrenheit sabendo-se que tem-se 30 graus Celsius. Sabendo-se que a formula é a seguinte

$$F = C * 1,8 + 32$$

, onde *F* são os graus Fahrenheit e *C* os graus Celsius.

1.3 Parte III - Extra Windows Forms e Web Forms

Exercício 21.

Construa um programa em windows forms que tenha um button e um textbox. Ao clicar no button deve aparecer a mensagem “Alo mundo” no textbox

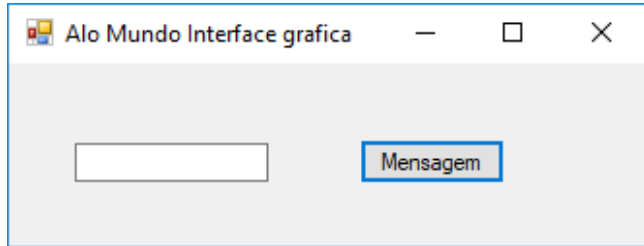


Figura 1. Alo Mundo antes de Clicar no Botão

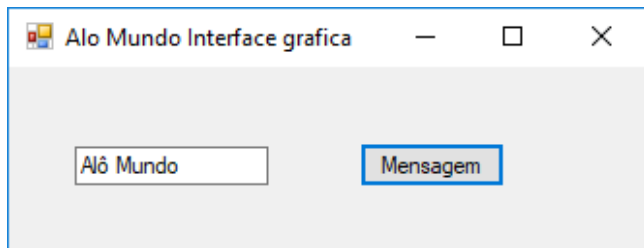


Figura 2. Alo Mundo depois de Clicar no Botão

Exercício 22.

Construa um programa em Web forms que tenha um button e um textbox. Ao clicar no button deve aparecer a mensagem “Alo mundo” no textbox

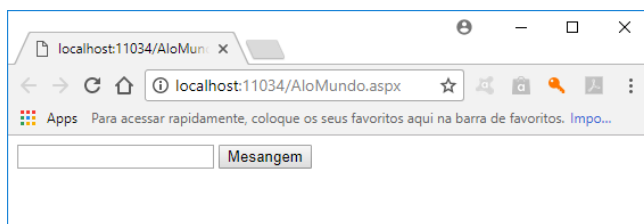


Figura 3. Alo Mundo antes de Clicar no Botão

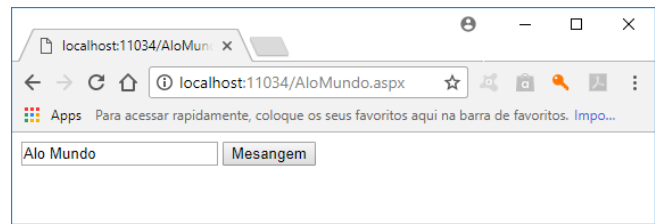


Figura 4. Alo Mundo depois de Clicar no Botão

- <http://miktex.org>
- <http://www.leg.ufpr.br/walmes/tikz/>

Referências

- [1] Adelaide Carvalho. *Prática de C# - Algoritmia e Programação Estruturada*. FCA - Editora de Informatica, Lda, Lisboa - Portugal, 1 edition, 2010.
- [2] Adelaide Carvalho. *Prática de C# - Programação Orientada Por Objetos*. FCA - Editora de Informática, Lda, Lisboa - Portugal, 1 edition, 2011.

Agradecimentos

Agradecemos a toda equipe que atua nos bastidores, sem os quais não conseguiríamos realizar este trabalho de promover a educação em nosso país.

Formatação deste documento

Para a confecção deste documento usaram-se os recursos obtidos gratuitamente nos sites citados a seguir:

- <http://www.latextemplates.com/template/stylish-article>
- <http://www.texniccenter.org/>