



Universidade Federal de Uberlândia  
Faculdade de Computação – Gestão da Informação – Prof. Daniel A. Furtado  
2º Trabalho de Programação Orientada a Objetos  
Revisão – Estruturas de Repetição e Arrays em C#

### Instruções Gerais

---

- Esta atividade deve ser realizada **individualmente**;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet, gerados por ferramentas de IA (como ChatGPT, Gemini e similares), copiados de trabalhos de semestres anteriores, serão anulados;
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados a eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

### Material de Apoio

---

<https://furtado.prof.ufu.br/site/teaching/POO/POO-Modulo1-Revisao.pdf> (slides 1-30)

**OBS:** Para agilizar o trabalho, não é necessário criar um novo projeto no Visual Studio para cada exercício. Assim que terminar um exercício e testar a solução, basta comentar o código (para não atrapalhar na solução do próximo), acrescentar o comentário “Exercício X”, e seguir adiante. Ao terminar todos eles, envie um único arquivo (trab2.cs) contendo todas as soluções, sem compactar.

### Exercício 1

---

Escreva um programa em C# que receba um inteiro **n** do usuário e mostre na tela todos os números **ímpares positivos** menores que **n**, em ordem **crecente**. Utilize a estrutura de repetição **for**. Não utilize a estrutura **if** nem o operador de resto de divisão **%**.

### Exercício 2

---

Escreva um programa em C# que receba um inteiro **n** do usuário e mostre na tela todos os números **pares positivos** menores que **n**, em ordem **decrecente**. Utilize a estrutura de repetição **for**. Não utilize a estrutura **if** nem o operador de resto de divisão **%**.

### Exercício 3

---

Escreva um programa em C# que receba uma string do usuário e contabilize o número de letras 'a' presentes na string. A verificação deve ser feita caractere por caractere da string utilizando a estrutura **for**. A quantidade encontrada deve ser apresentada ao usuário.

### Exercício 4

---

Escreva um programa em C# que receba uma string do usuário e contabilize o número de caracteres da string que **não** são dígitos. Por exemplo, se o usuário informar a string “ab234c1”, o

programa deve apresentar a resposta **3** (pois há três caracteres não dígitos: **a**, **b** e **c**). Utilize a estrutura **for** para acessar cada caractere da string. Para verificar se um caractere corresponde a um dígito, utilize `Char.IsDigit` (exemplo: `if (Char.IsDigit(minhaString[i]))...`).

## Exercício 5

---

Escreva um programa em C# que receba 10 strings do usuário e as armazene em um **array** de strings de 10 posições. Depois de solicitar todas as strings o programa deverá apresentá-las na ordem em que foram fornecidas utilizando a estrutura **foreach**, e na ordem contrária utilizando a estrutura **for**. O programa deve ter o menor número possível de linhas.

## Exercício 6

---

Escreva um programa em C# que solicite o **nome** do usuário. Caso o nome informado tenha menos de 3 caracteres, o programa deve informar que tal nome não é aceito e repetir a solicitação até que o usuário informe um nome válido (com 3 ou mais caracteres). Utilize a estrutura de repetição **while**.

## Exercício 7

---

Escreva um programa em C# que receba números do usuário, um de cada vez, e os armazene em um **array** de **inteiros**. O programa deve parar de solicitar os números quando o usuário pressionar **enter** sem digitar nenhum valor (string vazia). No final do processo, o programa deve apresentar todos os números informados, na ordem contrária, e o somatório deles. O **array** deve ser criado para armazenar até 100 números. Não utilize listas.

## Exercício 8

---

Escreva um programa que apresente ao usuário o seguinte menu de opções:

- 1 – Calcular a área de um círculo
- 2 – Calcular a área de um retângulo
- 3 – Sair

O programa deve solicitar a opção desejada ao usuário e, em seguida, utilizando a estrutura **switch-case**, solicitar os dados para cálculo e apresentação da respectiva área. O programa deve reapresentar o menu de opções para que o usuário possa escolher outra opção e fazer um novo cálculo, se ele desejar. O programa deve finalizar apenas quando o usuário escolher a opção **3 – Sair**. Utilize a estrutura **do-while**.

## Sobre Eventuais Plágios

---

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.