EXERCICIO COMPLEMENTARES: UNIDADE 8

Resumo do Capítulo

* Os programas de computador utilizam as instruções de decisão para definir se um bloco de código será executado ou não de acordo com determinada condição.
* Para utilizar a instrução de decisão **if**, devemos definir uma condição e um corpo. O corpo é um bloco de código executado somente se a condição for **true**.
* Para utilizar a instrução **else**, devemos definir um corpo. Essa instrução sempre está relacionada à instrução **if**. O corpo do **else** é executado somente se a condição do **if** correspondente for false.
* Instruções de decisão podem ser encadeadas. Em outras palavras, podemos definir **ifs** e **elses** dentro de **ifs** ou **elses**.
* Os programas de computador utilizam as instruções de repetição para executar repetidas vezes um determinado bloco de código.
* Para utilizar a instrução de repetição **while**, devemos definir uma condição e um corpo. O corpo é executado enquanto a condição for **true**. Toda vez, depois que o corpo é executado, a condição é reavaliada para decidir se o laço deve ou não continuar.
* Para utilizar a instrução de repetição for, devemos definir uma inicialização, uma condição, uma atualização e um corpo. O fluxo de execução começa pela inicialização e, enquanto a condição for true, ele executa de forma cíclica a condição, o corpo e a atualização nessa ordem.
* Instruções de repetição podem ser encadeadas. Em outras palavras, podemos definir whiles e fors dentro de whiles ou fors.
* A instrução break interrompe a execução de um laço.
* A instrução continue interrompe a execução de uma iteração. No while, o continue desvia o fluxo de execução para a condição. No for, o continue desvia o fluxo de execução para a atualização.
* Quando o corpo do if possui apenas um comando, ele não precisa ser delimitado com chaves.
* A mesma regra vale para o else, while e for.
* Se a condição de umlaço sempre é true, o corpo desse laço será executado repetidamente sem parar(“laços infinitos”).

**1)** Crie um programa em C# que gere dois números aleatórios e exiba o valor desses números. Além disso, esse programa deve exibir a mensagem “Primeiro > Segundo” se o primeiro número for maior do que o segundo, a mensagem “Segundo > Primeiro” se o segundo número for maior do que o primeiro e mensagem “Primeiro = Segundo” se o primeiro número for igual ao segundo.

**2)** Crie um programa em C# que exiba o seguinte padrão no terminal.

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

**3)** Para controlar o estacionamento de um condomínio, devemos implementar um programa em C# para gerar os cartões das vagas dos moradores. Nos cartões, é necessário constar o número do bloco e o número do apartamento. Nesse condômino, há três blocos numerados de 1 a 3. Em cada bloco, há 9 andares. Em cada andar, há 4 apartamentos. No primeiro andar, os números dos

Apartamentos são: 11, 12, 13 e 14. No segundo andar, os números dos apartamentos são: 21, 22, 23 e 24. Nos outros andares, a lógica de numeração é a mesma.

**4)** Escreva um programa em C# que exiba no terminal a tabuada dos números de 1 a 10 de acordo com o padrão abaixo.

1x1 = 1

1x2 = 2

1x3 = 3

...

10x8 = 80

10x9 = 90

10x10 = 100

**5)** Considere um jogo no qual o jogador lança um dado 10 vezes. O jogador ganha se a soma dos valores obtidos nos lançamentos for ímpar. Se o número 6 for sorteado 2 vezes o jogador perde imediatamente. O número 1 deve ser desconsiderado na somatória. Implemente um programa em C# para simular a execução desse jogo

**6)** Crie um programa em C# que gere e exiba no terminal um número aleatório. Além disso, esse programa deve contabilizar a quantidade de dígitos “1” do número aleatório e exibir essa quantidade no terminal. Complete o código abaixo.

Bons estudos!