

UPskill – Java+.NET

Programação Orientada a Objetos – Classes e Objetos

EXERCÍCIO SISTEMA DE INFORMAÇÃO DE ESTUDANTES

1. Crie um programa em C# para um Sistema de Informação de Estudantes simples. Você precisa criar uma classe `Student` com as seguintes propriedades e métodos:

- Propriedades:
 - `StudentId` (int)
 - `FirstName` (string)
 - `LastName` (string)
 - `Age` (int)
 - `Grade` (double)
 - `IsActive` (boolean)
 - `Course` (string)
- Construtores:
 - Um construtor padrão.
 - Um *overload* ao construtor que permite ao utilizador instanciar com `StudentID`, `FirstName`, `LastName`, `Age` e `Grade`.
- Propriedades e métodos estáticos:
 - Uma propriedade estática `StudentCount` para manter o registo do número total de estudantes.
 - Um método estático `GetStudentCount` para obter o valor de `StudentCount`.
- Métodos públicos:
 - `GetFullName()` para devolver o nome completo do aluno.
 - `SetCourse(string course)` para definir o curso do aluno.

No método principal (`main`), permita que o utilizador introduza dados relativos a pelo menos dois alunos (utilize um ciclo) e, em seguida, apresente a informação sobre os alunos, incluindo o número total de alunos.

EXERCÍCIO SISTEMA DE GESTÃO DE COMBUSTÍVEL DE AUTOMÓVEIS

1. Crie um programa C# para um Sistema de Gestão de Combustível de Automóveis. É necessário criar uma única classe Carro com propriedades e métodos para gerir os dados de entrada de combustível e calcular relatórios de consumo mensal e médio:

- Propriedades:
 - Make (string) - a marca do automóvel.
 - Model (string) - o modelo do automóvel.
 - Year (int) - o ano do automóvel.
 - FuelConsumption (double[]) - uma matriz para armazenar a quantidade de gasolina consumida em cada mês.
- Construtores:
 - Um construtor que inicializa a Marca, o Modelo e o Ano do automóvel.
- Métodos públicos:
 - AddFuelInput(double gallons, int month) para adicionar uma entrada de gasolina para a mês específico.
 - CalculateMonthlyReports() para calcular e devolver os relatórios mensais de consumo de combustível.
 - CalculateAverageConsumption() para calcular e devolver o consumo médio de combustível do ano.
 - DisplayMonthlyReports() para mostrar os relatórios mensais de consumo de combustível, incluindo o total de combustível consumido em cada mês e o consumo médio do ano.

No método principal (main), permita ao utilizador criar um automóvel, adicione entradas de combustível para cada mês e apresentar os relatórios mensais, incluindo o consumo médio do ano.

EXERCÍCIO FORMAS GEOMÉTRICAS COM VERIFICAÇÃO DE SOBREPOSIÇÃO DE RETÂNGULOS

1. Crie um programa em C# que permita a introdução de formas geométricas e permita verificar se um retângulo se sobrepõe a outro. Considere:

- Classe Point
 - Propriedades: X (int) e Y (int) para representar as coordenadas.
- Classe Circle
 - Propriedades: Center (Point) para representar o centro da circunferência, e Radius (int) para representar o raio.
 - Métodos: CalculateArea() para calcular e retornar a área do círculo, e CalculatePerimeter() para calcular e retornar a circunferência do círculo.
- Classe Rectangle
 - Propriedades: TopLeft (Point) para representar o canto superior esquerdo, Width (double) e Height (double) para representar as dimensões.
 - Métodos: CalculateArea() para calcular e devolver a área do retângulo, CalculatePerimeter() para calcular e devolver o perímetro do retângulo, e OverlapsWith(Rectangle otherRectangle) para verificar se o retângulo atual se sobrepõe a outro retângulo.
- Classe Triangle
 - Propriedades: Ponto1, Ponto2 e Ponto3 (Point) para representar os três vértices do triângulo.
 - Métodos: CalculateArea() para calcular e retornar a área do triângulo, e CalculatePerimeter() para calcular e retornar o perímetro do triângulo.

No método principal (main), crie várias instâncias destas classes e realize várias operações geométricas, incluindo verificar se os retângulos se sobrepõem. Apresente os resultados ao utilizador.