RE qualifica-te

Ficha 26
Programação C#
Interfaces

Rua Flores de Lima Nº16, Lisboa
cv@recodme.pt

INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL
Lisb@20[20] PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

Cada vez que realizares uma ficha que reutiliza algum código, faz uma cópia do código reutilizado em vez de alterares as respostas originais. Documenta corretamente todo o código que produzires

Grupo I – Interfaces como contratos comportamentais

1. Cria uma interface chamada IVehicle que deverá conter as assinaturas para os métodos Start, Stop, ChangeGear, Break, Accelerate e TurnWheel.

2. Cria uma classe que identifique uma mota

3. Cria uma classe que identifique um barco

4. Implementa a interface em todas as classes anteriores. Implementa a classe na classe criada na ficha 24.

5. Cria a interface IErasable com o método erase.

6. Cria a interface IWritingUtensil com o método write e a propriedade color.

7. Cria as classes Pen, Pencil, ColoredPencil. Implementa as interfaces que considerares corretas. Elabora as classes com propriedades, métodos e construtores que achares necessários.

8. Paper e PaperContent em anexo permitem, através de um IWritingUtensil, escrever texto. Poderá ser possível apagar o último texto introduzido caso seja possível. Analisa o código, compreende o que faz, e testa a classe para validares o que foi feito nos exercícios 5, 6 e 7. Se considerares útil, escreve uma breve descrição do que a classe Paper faz.

Grupo II – Interfaces como tipos e o seu uso em genéricos

1. Cria uma interface chamada ITrashable
2. Implementa a interface nas classes desenvolvidas até agora, que quiseres.
3. Cria uma lista de objetos (List<object>)
4. Coloca nesta lista objetos das classes que criaste (pelo menos 1 de cada)
5. Cria uma função que receba uma lista de objetos como argumento. Esta função deverá iterar sobre cada um dos objetos e remover os que podem ser deitados fora (ITrashable).
6. Cria um método de extensão que faça o mesmo que a função acima descrita.

Grupo III – Indexadores

1. Cria uma classe chamada Movel. Nesta classe cria um indexador. Este indexador deverá ter a chave do tipo string, e o valor do tipo List<object>. Cada índex irá simular uma gaveta.
2. Cria um novo objeto do tipo Movel e cria um elemento no indexador com a chave “Tralha”.
3. Adiciona as gavetas que quiseres
4. Cria um método que limpe todas as gavetas. (ver Grupo II-5)
5. Cria um método que organize o conteúdo das gavetas. O que não tem gaveta, vai para a gaveta da tralha.
6. [Avançado] Faz o mesmo, mas agora com a chave do tipo Type. Podes criar interfaces que te permitam arrumar coisas semelhantes numa gaveta. Caso possa ser deitado fora, coloca-o na gaveta da tralha.