

Programação Visual

Trabalho de Laboratório nº 0

Objetivo

Familiarização com o ambiente de desenvolvimento Visual Studio.
Criação de aplicações de consola simples.
Introdução/Conversão à linguagem C# com a criação/utilização de classes, herança, polimorfismo e interfaces.

Programa

Protótipo do Jogo de Xadrez.

Regras

Criar uma aplicação de consola (.Net Core).
Use as convenções de codificação adotadas para a linguagem C#.
Na classe do programa não coloque atributos nem crie nenhum método para além do **Main**.
Não é necessário obter dados do utilizador. Forneça os dados ao nível do código.

Descrição



Nível 1

- Comece por criar a classe **Posicao** que representa uma posição do tabuleiro de Xadrez. Esta classe tem os atributos **x** e **y**. O primeiro representa a coordenada **x** e guarda um caracter (entre 'a' e 'h'), o segundo a coordenada **y** e guarda um valor inteiro (entre 1 e 8). Complete a classe com dois construtores, um sem parâmetros que cria um objeto nas coordenadas **a** e **1** (use constantes de classe para guardar estes valores) e o outro com dois parâmetros que recebem o valor dos atributos. Inclua o método **Tostring()** que retorna um texto com os valores de **x** e **y** concatenados (ex: "a2").
- Teste a classe criada criando dois objetos usando construtores diferentes. Escreva no ecrã a informação dos objetos criados.

Nível 2

- Construa uma hierarquia de classes com base na classe *abstrata* **Peca** que inclui as classes derivadas **Peao** e **Torre**. A classe **Peca** deve guardar a informação da sua posição e da sua cor. Para a cor use a variável lógica **corBranca** que quando está a **true** significa que a peça é branca e quando está a **false** que é preta. Inclua os construtores, um sem parâmetros e o outro com dois parâmetros que recebem o valor dos atributos, os métodos modificadores e seletores e o método **Tostring()** que retorna a posição da peça usando o método equivalente da classe **Posicao**.
- As classes derivadas **Peao** e **Torre** não acrescentam atributos e ambas têm apenas um construtor que recebe a cor e a posição da peça. O **Tostring()** da classe **Peao** devolve o mesmo que o **Tostring()** da classe **Peca**. O **Tostring()** da classe **Torre** acrescenta antes da posição a letra T maiúscula que representa a torre (ex:Ta8).
- Teste as classes criadas.

Programação Visual

Trabalho de Laboratório nº 0

Nível 3

- Inclua agora um método **GetNome()** na classe **Peca** que retorna o texto “desconhecida”. Redefina este método nas classe **Peao** e **Torre** de forma a retornar o texto “Peão” e “Torre” respetivamente.
- Para testar o polimorfismo crie um *array* de objetos da classe **Peca** e coloque no mesmo pelo menos um objeto da classe **Peao** e outro da classe **Torre**. Pode usar os objetos que criou antes. De seguida, usando um ciclo **foreach**, percorra o *array* escrevendo no ecrã o nome de todas as peças que contem.

Nível 4

- Acrescente uma interface **IMover** para reunir as funcionalidades de movimento das peças. O único método a definir - **void Deslocar(int dx, int dy)** - deverá mover uma peça um determinado valor no eixo do x (**dx**) e outro no eixo do y (**dy**).
- Implemente a interface **IMover** na classe **Peca**.
Nota: Por agora não precisa ter em conta valores inválidos para o movimento, assuma apenas que o peão não se move na horizontal e a torre só aceita um dos movimentos (**dx** ou **dy**), se forem ambos diferentes de zero não se move.
- Teste.

Nível 5

- Redefina o operador incrementar (**++**) de forma a que possa ser usado pela classe **Peao** para o seu movimento. (ex: **Peao p1 = new Peao(); p1++;**).
- Teste.

Desafio

- Crie as classes que faltam para as peças de xadrez: **Cavalo**, **Bispo**, **Rei** e **Rainha**. Não coloque código nos métodos **Deslocar** destas classes.
- Crie a classe **Tabuleiro** que usa um *array* bidimensional para guardar as peças. Implemente nesta classe o construtor sem parâmetros que inicializa o tabuleiro com as peças normais do xadrez nas suas posições iniciais.
- Crie um método **Mostrar()** na classe **Tabuleiro** que escreve para o ecrã o tabuleiro com as suas peças. Use os símbolos das peças para as representar e inclua as coordenadas.
- Teste.

Notas

- Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:
- A notação camelCase para o nome das variáveis locais e identificadores privados.
 - A notação PascalCase para os nomes públicos dos métodos, classes e interfaces.
 - Não utilize o símbolo ‘_’ nos identificadores nem abreviaturas