

## PROGRAMAÇÃO ORIENTADA A OBJETOS I – SISTEMAS DE INFORMAÇÃO

### Lista de exercícios 6 – Herança

Prof. Jean Eduardo Glazar

#### EXERCÍCIO 1:

Consideremos uma simulação simples de um navio em um oceano. Este navio é representado por seu **nome** e está em uma determinada posição **(x,y)** no espaço de duas dimensões: pense na posição do navio como sendo um dos quadrados mostrados na figura abaixo, de forma análoga à de um jogo de Batalha Naval. Vamos considerar que as posições que um navio pode ocupar são *discretas*, ou seja, um navio não poderá ocupar “meia” posição ou uma posição cujo valor não seja inteiro. Um navio também tem uma **direção**, ou seja, a posição para onde está apontando, que pode ser Norte, Sul, Leste ou Oeste (**N,S,L,O**). Na figura, o navio **A** está nas coordenadas (5,7) e aponta para o norte; e o navio **B** está nas coordenadas (3,4) e aponta para leste.

		Y								
		1	2	3	4	5	6	7	8	9
X	1									
	2									
	3				B→					
	4									
	5							A↑		

- Implemente a classe **Navio** com um método para **modificar a direção** do navio e outro método para **mover** o navio uma posição. É necessário verificar qual a direção do navio antes de mover. Por exemplo, se ele estiver apontando para o norte, basta fazer  $x=x-1$ . Se ele estiver apontando para o leste, basta fazer  $y=y+1$ , e assim por diante.
- Considere também um **Navio Cargueiro**, que além das características acima, possui uma **capacidade máxima** e a **capacidade atual** de carga. Crie esta classe, com métodos para alterar a capacidade atual e a capacidade máxima.
- Considere agora um **Navio de Guerra**, que além de ter as capacidades de carga de um navio cargueiro, deve ter o **número de canhões** e o **número de soldados**. Escreva a classe **Navio de Guerra** usando as classes desenvolvidas até agora, com um método para alterar a quantidade de soldados e outro para alterar o número de canhões.
- No programa principal, crie 3 objetos: um do tipo **Navio**, outro do tipo **NavioCargueiro** e o outro do tipo **NavioGuerra**. Faça todos se moverem, altere as informações de cada um e imprima novamente.

**OBSERVAÇÃO:** Todas as classes devem ter construtores e um método para imprimir todos os dados de cada classe.