

CC3642

Orientação a Objetos

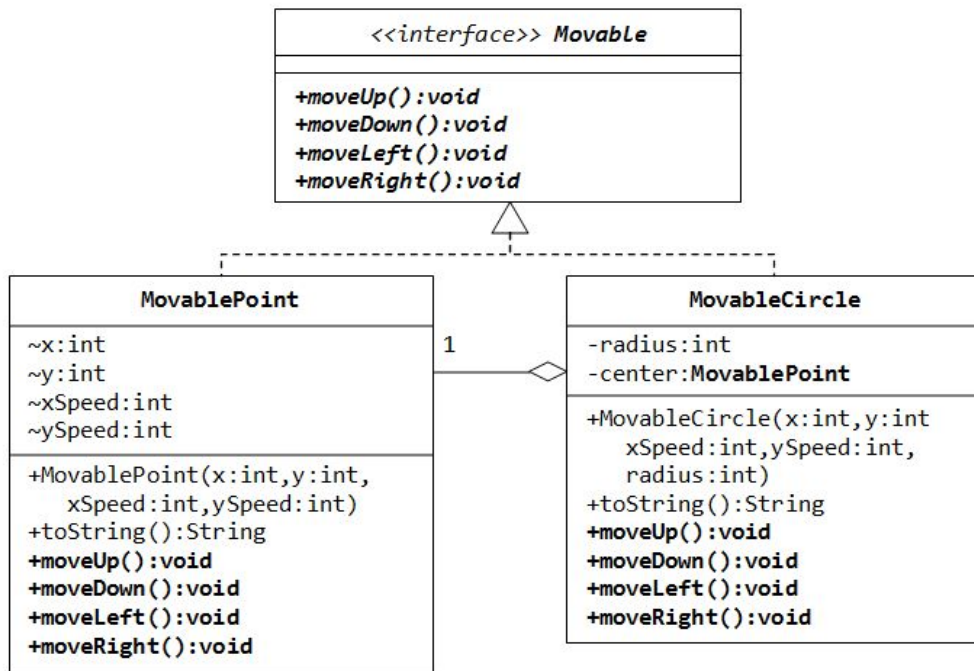
Prof. Danilo H. Perico

Exercício 1 - Sobrecarga de Método

Crie uma classe chamada *Area* para imprimir a área de um quadrado e de um retângulo. A classe deve ter, portanto, dois métodos com o mesmo nome, mas diferentes números de parâmetros. O método para imprimir a área do retângulo tem dois parâmetros que são comprimento e largura, respectivamente, enquanto o outro método para imprimir a área do quadrado tem somente um parâmetro que é o lado do quadrado.

Exercício 2

Implemente o Diagrama de Classes apresentado abaixo.



Modificador de acesso ~ (til): Os atributos ou métodos podem ser acessados por qualquer objeto dentro do pacote

No Java, basta não colocar nada: package-private

Exercício 3 - *Matrix* (Saraiva Jr., O. - Cap. 9)

“Neo, a Matrix é um grande sistema orientado a objetos, em que as pessoas são objetos de classes que herdam da classe abstrata Agente.”

Dada a definição acima de *Matrix*, implemente as classes *Empresario*, *Professor* e *Advogado* como subclasses da classe *Agente*:

- A classe *Agente* tem os seguintes atributos: String nome; boolean modo_agente; String profissao; o método abstrato void apresentacao() e o método modo_agente_on();
- A classe *Empresario* deve ter os atributos: String nome; boolean modo_agente; String profissao, String empresa; e o método void apresentacao();
- A classe *Professor* deve ter os atributos: String nome; boolean modo_agente; String profissao, String escola; e o método void apresentacao();
- A classe *Advogado* deve ter os atributos: String nome; boolean modo_agente; String profissao, String OAB; e o método void apresentacao();

Exercício 3 - *Matrix*

No método *main()* a explicação de *Matrix* deve ser implementada.

Um ArrayList de n pessoas do tipo Agente deve ser definido. Então, com um loop infinito, faça um menu para criar pessoas com diferentes profissões dentro deste ArrayList.

Coloque uma opção no menu para todas as pessoas se apresentarem.

Escolha algumas pessoas para invocar o método *modo_agente_on()* e transformá-las em Agentes. Cuidado: os objetos não podem ser instanciados como Agentes (classe abstrata), mas inclua um print no método *apresentacao()* que leia o atributo *modo_agente* e imprima “AGENTE SMITH”, para *modo_agente = true*.

Exercício 4

Implemente o Diagrama de Classes ao lado. Crie também uma classe para testar o funcionamento de objetos da classe Morcego.

