
Curso	Sistemas de Informação
Disciplina	Programação Orientada a Objetos 1
Professor	José Gustavo de Souza Paiva

1. Tema central

Criação de classes com utilização de construtores.

2. Conteúdo

Para cada um dos problemas abaixo, construa o programa completo na linguagem de programação Java capaz de verificar a funcionalidade de cada uma das funções solicitadas.

- a) Crie um programa que implemente uma máquina de passagens, seguindo as ideias apresentadas na aula teórica.
- b) Crie uma classe **Tempo** com os atributos **hora**, **minuto** e **segundo**. A classe **Tempo** deve prover as seguintes funcionalidades:
- Os objetos devem poder ser criados com os atributos já configurados, ou seja, é possível criar um objeto passando todas as informações. A classe só deve permitir associar valores válidos para os seus atributos (hora (0..23), minuto (0..59), e segundo (0..59)). Caso os valores passados sejam inválidos, deve-se atribuir 0 para o respectivo valor;
 - A classe deve permitir mostrar o tempo no formato "hhmmss";
 - A classe deve permitir mostrar o tempo no formato "hh:mm:ss AM (ou PM)";
 - Faça dois métodos que simulem os botões de ajuste de minutos, ou seja, quando esses métodos forem chamados, o valor do minuto deve ser incrementado/decrementado de uma unidade. Esses métodos devem garantir que a hora seja sempre válida, ou seja, caso o valor dos minutos seja alterado de 59 para 00, o valor da hora deverá ser adicionado de 1, e assim por diante.
 - Faça um programa principal que permita a criação de um objeto da classe tempo. Faça diversos testes para verificar as funcionalidades criadas. Sugira e implemente novos métodos/atributos para incrementar a classe.
- c) (Exercicio adaptado de http://www.feg.unesp.br/~gsena/Disciplinas/Turmas_2011/PC-II_EngenhariaMecanica/Aula_8/Aula_2/Lista2.pdf) Criar uma classe **NaveEspacial** tal como especificado no diagrama UML abaixo:

NaveEspacial	
-String	nome
-int	velocidade
+ NaveEspacial(String s)	
+ void mostrarNave()	

- Elabore um construtor que empregue o parâmetro *s* como valor para o campo *nome* de um objeto *NaveEspacial*. Além disso, o campo *velocidade* deverá ter um valor aleatório inteiro contido no intervalo [5, 10] (pesquise como implementar valores aleatórios em Java).
- Crie uma classe **AmbienteEspacial** que apresenta um menu com as opções de criar uma nova nave, e exibir as naves existentes. O sistema deve permitir a criação de até 5 objetos *NaveEspacial* (utilize um vetor para armazenar esses objetos). Crie um método **mostrarNave()** que exiba o nome da nave e o valor da velocidade. Se esse método for chamado para um objeto *Nave* cujo campo *velocidade* é igual a zero, então, deverá mostrar a mensagem "Nave inoperante". O sistema deve possuir uma funcionalidade para parar uma nave, no qual são exibidas todas as naves em movimento para o usuário escolher. Assim que ele escolhe, o sistema faz com que a velocidade da nave seja alterada para 0. Da mesma forma, crie uma funcionalidade para ligar uma nave, que funciona da mesma forma que a funcionalidade de parar, mas atribuindo sua velocidade com um valor escolhido. Se mais de 3 objetos do tipo *NaveEspacial* estiverem operando (velocidade diferente de 0), ao tentar "ligar" uma nave a mensagem "Superpopulação de Naves" deverá ser exibida junto ao menu principal, e a velocidade de nenhuma nave é alterada.
- Implementar o método **naveMaisRapida(NaveEspacial n)** que verifica se a nave passada como parâmetro possui maior valor de velocidade do que a nave invocadora do método.