
Trabalho 03 (Parte da AP1) — Semana 21/10 a 28/10
QXD0007 — Programação Orientada a Objetos — 2021.2

Criação de Classes e Aplicações em Java

Professor: Atílio Gomes Luiz

1 Questões

- (a) Escreva em Java a classe `NumeroComplexo` que represente um número complexo. A classe deverá ter os seguintes métodos:
 - `inicializaNumero`, que recebe dois valores como argumentos para inicializar os campos da classe (parte real e imaginária);
 - `imprimeNumero`, que deve imprimir o número complexo encapsulado usando a notação $a + bi$ onde a é a parte real e b a imaginária;
 - `ehIgual`, que recebe outra instância da classe `NumeroComplexo` e retorna `true` se os valores dos campos encapsulados forem iguais aos da instância passada como argumento;
 - `soma`, que recebe outra instância da classe `NumeroComplexo` e soma este número complexo com o encapsulado usando a fórmula $(a + bi) + (c + di) = (a + c) + (b + d)i$;
 - `subtrai`, que recebe outra instância da classe `NumeroComplexo` e subtrai o argumento do número complexo encapsulado usando a fórmula $(a + bi) - (c + di) = (a - c) + (b - d)i$;
 - `multiplica`, que recebe outra instância da classe `NumeroComplexo` e multiplica este número complexo com o encapsulado usando a fórmula $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$;
 - `divide`, que recebe outra instância da classe `NumeroComplexo` e divide o número encapsulado pelo passado como argumento usando a fórmula $\frac{a+bi}{c+di} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$.
- (b) Escreva uma aplicação que demonstre o uso de instâncias da classe `NumeroComplexo`, criado no item acima. Demonstre o uso de todas as operações.

2. (a) Escreva uma classe `Data` cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

- **construtor**: define a data de determinado objeto (através de parâmetro). Este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001.
- **compara**: recebe como parâmetro um outro objeto da Classe data, compara com a data corrente e retorna:
 - 0 se as datas forem iguais;
 - 1 se a data corrente for maior que a do parâmetro;
 - -1 se a data do parâmetro for maior que a corrente.
- **getDia**: retorna o dia da data.
- **getMes**: retorna o mês da data.
- **getMesExtenso**: retorna o mês da data corrente por extenso.
- **getAno**: retorna o ano da data.
- **isBissexto**: retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário.
- **clone**: o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Data com os mesmos valores de atributos e retorna sua referência pelo método.

(b) Escreva uma aplicação que demonstre o uso de instâncias da classe `Data`, criado no item acima. Demonstre o uso de todas as operações.

3. (a) Escreva uma classe em que cada objeto representa um voo que acontece em determinada data e em determinado horário. Cada voo possui no máximo 100 passageiros, e a classe permite controlar a ocupação das vagas. A classe deve ter os seguintes métodos:
- **construtor**: configura os dados do voo (recebidos como parâmetro): número do voo, data (para armazenar a data utilize um objeto da classe Data, criada na questão anterior);
 - **proximoLivre**: retorna o número da próxima cadeira livre;
 - **verifica**: verifica se a cadeira, cujo número recebido como parâmetro, está ocupada;
 - **ocupa**: ocupa determinada cadeira do voo, cujo número é recebido como parâmetro, e retorna verdadeiro (operação foi bem sucedida) se a cadeira ainda não estiver ocupada e falso caso contrário;
 - **vagas**: retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo;
 - **getVoo**: retorna o número do voo.
 - **getData**: retorna a data do voo (na forma de objeto)
 - **clone**: o objeto clona a si próprio. Para isto, ele cria um novo objeto da mesma classe e faz uma cópia dos valores de seus atributos.
- (b) Escreva uma aplicação que demonstre o uso de instâncias da classe VOO, criado no item acima. Demonstre o uso de todas as operações.