UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Bacharelado em Sistemas de Informação
Técnicas de Programação
Prof. Paulo Sérgio Santos

7ª Lista de Exercícios
ArrayList, Generics e Classes Wrapper

Crie ou altere as classes conforme definido nos itens abaixo e crie um programa para testar essas classes:

1. Crie a classe `Equacao2Grau` que possui os atributos `a`, `b` e `c` da expressão

   $$ax^2 + bx + c = 0$$

   e implemente os métodos

   - `Float getX1()`
   - `Float getX2()`

   que devem retornar:

   $$x1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

   $$x2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

   Dicas: use o método estático *Math.sqrt* para calcular a raiz quadrada. Cuidado: não existe raiz quadrada de valor negativo. Nesse caso é impossível calcular os valores de `x1` e `x2` e os métodos devem retornar `null` !

2. Ler duas listas de inteiros positivos **a** e **b** (sem repetições) e gerar uma terceira lista **c** com a união de **a** e **b**. A lista **c** não pode ter números duplicados. Imprima a lista **c**.

3. Ler duas listas de inteiros positivos **a** e **b** e gerar uma terceira lista **c** com a interseção de **a** e **b**. A lista **c** não pode ter números duplicados. Imprima a lista **c**.

4. Ler duas listas de inteiros positivos **a** e **b** (sem repetições) e gerar uma terceira lista **c** com os elementos de **a** que não estão em **b**. Imprima a lista **c**.

5. Crie uma classe Aluno com Nome, nota A1 e nota A2. Faça um programa para ler o nome, a nota A1 e a nota A2 de uma lista de alunos, até que o usuário entre com um nome vazio. Ao final imprima o nome do aluno, as notas A1 e A2 e a média. Caso a média seja maior que 6, imprima APROVADO, senão imprima REPROVADO. Atente para o fato que nem todo aluno tem as notas A1 e A2 (o aluno pode ter faltado à prova). Nesse caso, o programa deve imprimir FALTOU na nota correspondente e REPROVADO.

6. Faça um programa para ler uma lista de palavras até que o usuário digite uma string vazia. Ao final, imprima a lista de palavras digitadas em ordem alfabética. Importante: não podem existir palavras duplicadas na lista.

7. Faça um programa para ler um número **n** e imprimir os n primeiros números da série de Fibonacci na ordem inversa. Lembre-se que a série de Fibonacci começa sempre com 0 e 1 e o próximo

número é calculado pela soma dos dois últimos. Dica: use um ArrayList para armazenar os números da série e depois imprimi-los na ordem inversa.

8. Em uma universidade:
   a) Todos os alunos são cadastrados com nome e matrícula.
   b) Cada disciplina possui código, nome, nº de créditos e valor mensal/crédito.
   c) Cada turma possui um código e está associada à disciplina que oferece.
   d) Uma turma pode ter, no máximo, 30 alunos inscritos.
   e) Os alunos podem se matricular em até 20 créditos.
   f) O valor da mensalidade do aluno é calculado somando o valor de cada disciplina na qual ele está matriculado.
   g) Existem alunos bolsistas que recebem um percentual de desconto no valor da mensalidade.

   Crie as classes necessárias para modelar o cenário acima e implemente métodos para:
   a) Matricular um aluno em uma turma
   b) Listar a disciplina e os alunos de uma turma
   c) Calcular o valor a ser pago mensalmente pelo aluno

   Importante: perceba que uma turma possui um conjunto de alunos e que um aluno possui um conjunto de turmas nas quais está inscrito

9. Faça um programa para manipular uma lista de números inteiros. O programa deverá ler os comandos definidos abaixo e gerar as saídas esperadas:
   a) i: insere um número na lista. Exemplo: "i 22" insere o número 22 na lista e imprime a mensagem "22 inserido".
   b) r: remove um número da lista. Exemplo: "r 15" remove o número 15 da lista e imprime a mensagem "15 removido". Se o 15 não fizer parte da lista deve ser impresso "15 não encontrado"
   c) p: imprime a lista de números ordenada crescentemente
   d) l: limpa a lista
   e) q: termina o programa

10. Cria uma classe para representar um polígono qualquer. Um polígono é composto por um conjunto de pontos e cada ponto possui duas coordenadas (x, y). A classe Poligono deve possuir métodos para adicionar e remover pontos e um método para calcular seu perímetro.

11. Crie uma classe `CarrinhoCompras` para armazenar os produtos implementados no exercício 33. Devem ser criados métodos para adicionar e remover produtos do carrinho. Repare que o cliente pode comprar várias quantidades do mesmo produto, ou seja, ele pode comprar 3 quantidades de um mesmo CD ou 2 quantidades de um mesmo livro. Implemente um método para alterar a quantidade de determinado produto do carrinho (se a quantidade for zero o produto deverá ser removido do carrinho). Implemente também um método para imprimir a lista de produtos no carrinho com descrição, preço, quantidade e o valor do item, além do valor total dos produtos, do valor total do frete e do valor total a pagar.

    a) boolean adicionar(Produto p)
    b) boolean remover(int codProduto)
    c) boolean alterar(codProduto, int quantidade)
    d) void imprimir()

12. Um dentista oferece três tipos de serviço: reparação, limpeza e orçamento. Crie uma agenda na qual seja possível marcar e desmarcar consultas. Cada consulta deve ter o serviço a ser realizado, o nome do paciente, a data e a hora de início e fim da consulta. Não é possível marcar uma consulta com horários coincidentes. Crie também uma opção para imprimir a agenda de uma data específica. Dica: use a classe Data e Hora implementadas anteriormente.

    a) boolean marcar(String nomePaciente, Data data, Hora horaIni, Hora horaFim)
    b) boolean desmarcar(String nomePaciente, Data data)

13. Um funcionário possui matrícula (int), nome e salário (float). Um departamento possui uma lista de funcionários. Crie métodos para adicionar um funcionário ao departamento e para imprimir a lista de funcionários do departamento ordenada por matrícula, nome ou salário (o usuário deve escolher que tipo de ordenação deseja).