Projeto e Construção de Sistemas

Lista de Exercícios 7 - Coleções

Exercício 1:
Faça um programa que permita o usuário entre um conjunto de números inteiros (quantidade não é conhecida a priori) e imprima a soma e a média desses números.

Exercício 2:
Faça um programa que permita o usuário entre um conjunto de números inteiros (quantidade não é conhecida a priori) e armazene esses números em um arquivo via serialização XML.

Exercício 3:
Faça um programa que leia o conjunto de números inteiros armazenados no exercício anterior e imprima a soma e a média desses números.

Exercício 4:
Faça um programa que leia um conjunto de números inteiros armazenando-os em um ArrayList (lista1) - faça com que o usuário entre no mínimo 6 números, podendo ser números repetidos - e:

- Crie uma nova lista (lista2) a partir dos 4 primeiros números entrados pelo usuário e imprima essa nova lista.
- Crie uma lista (lista 3) a partir da lista entrada pelo usuário apenas com os números entre 1 e 10 e imprima o resultado.
- Remova todos os números pares (entre 2 e 10) da lista 2 e imprima o resultado.
- Multiplique os dois últimos números da lista 2 por 10 e armazene o resultado em lugar dos números originais.
- Peça para o usuário entrar um número e imprima se alguma das três listas criadas anteriormente contém o número.
- Imprima quais listas possuem os números 2, 4 e 5 (apenas se todos os três números fizerem parte da lista). Imprima também em quais posições cada um desses números aparecem na lista.
- Ordene os números em cada lista e imprima o resultado.
- Modifique todas as ocorrências do número 10 nas três listas pelo número 20 e imprima o resultado (teste o programa entrando pelo menos uma vez o número 10).

Exercício 5:
Faça um programa que leia um conjunto de números inteiros armazenando-os em um HashSet (set1) - faça com que o usuário entre no mínimo 6 números, podendo ser números repetidos - e:

- Crie um hashset (set2) a partir da lista entrada pelo usuário apenas com os números entre 1 e 10 e imprima o resultado.
- Remova todos os números pares (entre 2 e 10) da do set2 e imprima o resultado.
- Peça para o usuário entrar um número e imprima se alguma dos conjuntos criados anteriormente contém o número.
- Imprima quais conjuntos possuem os números 2, 4 e 5 (apenas se todos os três números fizerem parte do conjunto). Imprima também em quais posições cada um desses números aparecem no conjunto.
- Ordene os números em cada conjunto e imprima o resultado.
- Modifique todas as ocorrências do número 10 nos conjuntos pelo número 20 e imprima o resultado (teste o programa entrando pelo menos uma vez o número 10).

Exercício 6:
Faça o mesmo programa do exercício 4, agora usando uma estrutura do tipo TreeSet.

Exercício 7:
Crie uma classe Produto com atributos nome, código e preço e faça um programa que crie algumas instâncias de produto e armazene-as em uma coleção do tipo HashSet. Agora, crie uma outra instância (vamos chama-la de prodAlvo) com os mesmos valores de atributos de um dos produtos armazenados no passo anterior. Faça uma chamada a um método do HashSet que verifique se o produto (prodAlvo) pertence à coleção criada.

Exercício 8:
Crie uma classe Produto com atributos nome, código e preço e faça um programa que crie algumas instâncias de produto e armazene-as em uma coleção do tipo TreeSet.
Imprima, em ordem crescente do nome do produto, o nome, o código e o preço dos produtos armazenados na coleção.
Agora imprime, em ordem decrescente do preço do produto as mesmas informações dos produtos.

Exercício 9:
Crie uma classe Produto com atributos nome (string), código (int) e preço (double) e faça um programa que crie algumas instâncias de produto e armazene-as em uma coleção do tipo HashMap, indexadas pelo código do produto. Portanto, teremos um mapeamento Integer->Produto.
Acrescente agora ao código a possibilidade do usuário entrar um código do produto e o programa, em resposta, deve dizer se o produto existe ou não e, caso exista, imprimir os dados do produto (nome e preço).
Acrescente agora ao código: o programa ao final deve imprimir a relação de todos os códigos de produtos armazenados na coleção.

Exercício 10:
Crie uma classe Produto com atributos nome (string), código (int) e preço (double) e faça um programa que crie algumas instâncias de produto e armazene-as em uma coleção do tipo TreeMap, indexadas pelo código do produto. Portanto, teremos um mapeamento Integer->Produto.
Acrescente agora ao código a possibilidade do usuário entrar um código do produto e o programa, em resposta, deve dizer se o produto existe ou não e, caso exista, imprimir os dados do produto (nome e preço).
Agora, além de imprimir os dados do produto, o produto deve mudar o código do produto encontrado, multiplicando-o por 10 e atualizar essa nova configuração do produto na coleção de forma apropriada.
Acrescente agora ao código: o programa ao final deve imprimir a relação de todos os códigos de produtos armazenados na coleção.

Exercício 11:
Crie uma classe Disciplina com a seguinte definição:
Atributos:

- código : String
- nome : String

Operações:

- Construtor recebendo código e nome
- get e set para os atributos código e nome.

Crie uma classe Aluno com a seguinte definição:
Atributos:

- nome : String
- disciplinas : conjunto de objetos da classe Disciplina (Set de Disciplina)
- notas: mapeamento <código disciplina, lista de notas>, ou seja, para cada código de disciplina (chave do mapa) existe uma lista de notas do aluno nessa disciplina.

Exemplo:
Nome: João
Disciplinas: [(T011, Disciplina-1), (T022, Disciplina-2)]
Notas: [(T011, [100, 90, 95]), (T022, [50, 80, 90, 100])]

Operações:

- Construtor recebendo nome
- Operação get/set para nome
- Operação adicionaDisciplina(Disciplina d) que adiciona a disciplina d ao conjunto de disciplinas do aluno.
- Operação removeDisciplina(Disciplina d) que remove a disciplina d do conjunto de disciplinas do aluno e remove a entrada relativa a essa disciplina da estrutura de dados das notas. Se o aluno não estiver inscrito na disciplina d, a operação deve arremessar uma exceção.
- Operação adicionaNota(Disciplina d, int nota) que adiciona a nota na lista referente ao código da disciplina d. A nota deve ser acrescentada na última

posição dessa lista. Se o aluno não estiver inscrito na disciplina d ou se a nota for < 0 ou > 100, a operação deve arremessar uma exceção.

- Operação mudaNota(Disciplina d, int numero, int novaNota) que modifica a nota existente na posição dada pelo parâmetro numero da lista de notas da disciplina para a nova nota passada como parâmetro. Se o aluno não estiver inscrito na disciplina d ou se não possuir a nota com número de ordem passado como parâmetro ou se a novaNota for < 0 ou > 100, a operação deve arremessar uma exceção.
- Operação removeNota(Disciplina d, int numero) que remove a nota na lista referente ao código da disciplina d presente na posição dada pelo parâmetro numero. Se o aluno não estiver inscrito na disciplina ou se não possuir a nota com número de ordem passado como parâmetro, a operação deve arremessar uma exceção.
- Operação media(Disciplina d) que retorna a média das notas do aluno na disciplina d. Se o aluno não possuir notas na disciplina d ou se ele não estiver inscrito na disciplina d, a operação deve arremessar uma exceção.
- Operação mediaGeral() que retorna a média das médias do aluno nas suas disciplinas. Se o aluno não estiver inscrito em nenhuma disciplina ou se não possuir notas em uma das disciplinas em que ele estiver inscrito, a operação deve arremessar uma exceção.

Faça uma outra classe (TesteAluno) que teste a implementação da classe Aluno, com o seguinte cenário de execução

Crie as disciplinas (D01 – PCS) e (D02 – ED1)
Crie os alunos Joana e Marcelo
Adicione a disciplina D01 a Joana
Adicione a disciplina D02 a Joana
Adicione a disciplina D01 a Marcelo
Adicione a primeira nota 90 a Joana em PCS
Adicione a segunda nota 80 a Joana em PCS
Adicione a primeira nota 70 a Joana em ED1
Obtenha a media de Joana em PCS (85)
Obtenha a media geral de Joana (77.5)
Mude a nota 90 de Joana em PCS para 95
Obtenha a media de Joana em PCS (87.5)
Obtenha a media geral de Joana (78.75)
Remova a primeira nota 70 de Joana em ED1
Adicione a primeira nota 60 a Joana em ED1
Adicione a segunda nota 90 a Joana em ED1
Obtenha a media de Joana em ED1 (75)
Obtenha a media geral de Joana (81.25)
Tente remover a disciplina D02 de Marcelo (deve receber uma exceção)
Tente adicionar a nota 100 na disciplina D02 a Marcelo (deve receber uma exceção)
Tente mudar a quinta nota de PCS da aluna Joana para 100 (deve receber uma exceção)

Tente mudar a segunda nota de PCS da aluna Joana para 200 (deve receber uma exceção)
Tente mudar a segunda nota de PCS da aluna Joana para -10 (deve receber uma exceção)
Tente remover a quinta nota de PCS da aluna Joana (deve receber uma exceção)
Tente obter a media de Marcelo na disciplina PCS (deve receber uma exceção)
Tente obter a media geral de Marcelo (deve receber uma exceção)
Adicione a disciplina D02 a Marcelo.
Tente obter a media de Marcelo na disciplina ED1 (deve receber uma exceção)
Tente obter a media geral de Marcelo (deve receber uma exceção)