Classe ComparadorContaCorrenteSaldo, que implementa a interface Comparator (do JAVA)

Atributos – sem atributos

Métodos:
Construtor padrão.
int compare(ContaCorrente o1, ContaCorrente o2)
Deve retornar 1 se o saldo de o1 for maior do que o saldo de o2.
Deve retornar -1 se o saldo de o1 for menor do que o saldo de o2.
Deve retornar 0 se o saldo de o1 for igual ao saldo de o2.

Classe MediatorContaCorrente

Atributos: daoContaCorrente (do tipo DAOContaCorrente). Uma instância deve ser atribuída ao atributo na sua própria declaração.

Métodos:
Construtor padrão.
String incluir(ContaCorrente conta)
Deve validar os dados da conta corrente recebida e, se algum dado estiver inválido, retornar mensagem pertinente. As seguintes validações devem ser feitas:

- numero: dever ser diferente de nulo e de branco, e ter tamanho mínimo de 5 caracteres e máximo de 8 caracteres.
- saldo: deve ser maior ou igual a zero.
- nomeCorrentista: dever ser diferente de nulo e de branco, e ter tamanho máximo de 60 caracteres.
- Se a conta corrente recebida for uma conta poupança, validar o percentual de bônus que deve ser maior ou igual a zero.

Se os dados estiverem válidos, incluir a conta corrente no mecanismo de persistência usando o DAO. O retorno false do incluir do DAO indica que a chave da conta corrente (o seu número) já existe. Se isto ocorrer, retornar mensagem pertinente.
Caso a inclusão seja realizada com sucesso, retornar null.
String creditar(double valor, String numero)
Deve validar os dados recebidos. se algum dado estiver inválido, retornar mensagem pertinente. As seguintes validações devem ser feitas:

- valor tem que ser maior ou igual a zero.
- numero tem que ser diferente de nulo e de branco.

O método deve buscar no DAO a conta corrente por número. Se ela não existir, retornar mensagem pertinente. Se ela existir, creditar, usando o método da conta corrente, valor na conta, alterar, no DAO, a conta, e retornar null.
String debitar(double valor, String numero)
Deve validar os dados recebidos. se algum dado estiver inválido, retornar mensagem pertinente. As seguintes validações devem ser feitas:

- valor tem que ser maior ou igual a zero.
- numero tem que ser diferente de nulo e de branco.

O método deve buscar no DAO a conta corrente por número. Se ela não existir, retornar mensagem pertinente. Se ela existir, verificar se o saldo atual é menor que valor, se for, retornar mensagem pertinente. Caso contrário, debitar, usando o método da conta corrente, valor na conta, alterar, no DAO, a conta, e retornar null.
ContaCorrente buscar(String numero)
Se numero for nulo ou branco, retornar null. Caso contrário, realizar busca por número no DAO e retornar o retorno desta busca.
List<ContaCorrente> gerarRelatorioGeral()
Implementar tal método seguindo o modelo implementado no exemplo de Produto. A lista de contas correntes retornada deve estar ordenada por saldo, em ordem crescente. Para isto, usar o ComparadorContaCorrenteSaldo, já implementado.