1 – Atributos (5 itens):

• Encapsulamento: todos os atributos de classes não estáticas devem ser private

• Declarar pelo menos um atributo do tipo String

• Algum atributo deve ser do tipo de uma enumeração criada para o sistema;

• Deve ter pelo menos um atributo ArrayList;.

• Deve haver alguma constante declarada e usada;

2 – Métodos (2 itens):

2.1. Algum método deve ter uma lista de argumentos variados.

Metodo imprimirExtrato na class principal

2.2. Deve contém pelo menos um método estático no programa. Certamente que ele precisa ser necessário e utilizado.

3 – Construtor: Pelo menos duas classes do projeto deve possuir um construtor declarado;

4 – Associação por Composição

\*Uso da classe Conta como atributo da classe Cliente

5 – Associação por Agregação

\*Atributo conta\_destino na classe Operação

6 - Herança (2 itens)

6.1. Deve declarar algum atributo final em alguma Superclasse;

6.2. Pelo menos um acoplamento dinâmico (acesso a método da superclasse);

7 – Pelo menos um de cada Polimorfismo (4 itens)

7.1. Polimorfismo de sobrecarga;

\*Construtores da Classe ContaCorrente (Um abre a conta já com saldo, o outro abre com o padrão R$ 0.0

7.2. Polimorfismo de coerção;

\* Na classe BancoPos, dentro da opção 1 do switch.

7.3. Polimorfismo de subtipagem;

\* Metodo sacar reescrito na classe ContaPoupança.

7.4. Polimorfismo paramétrico;

\* Classe genérica Fundo e classes RendaFixa e RendaVariavel

8 – Pelo menos três classes do sistema devem implementar uma interface. Uma destas classes deve implementar, pelo menos, duas interfaces.

9 – Pacotes. Criar pelo menos dois pacotes.

\*Telas e Bancopos

10 – Classes abstratas. Pelo menos duas. Certamente que as concretas também devem ser implementadas

11 – Uma exceção deve ser criada, levantada e tratada pelo programa

\*Exceção criada no método sacar da classe conta e levantada no método sacar da Conta Corrente

12 – Deve ter alguma entrada de dados usando Caixas de Diálogo

13 – Alguma informação do sistema deve ser guardada em um arquivo e depois recuperada e usada.