

ALUNO: _____ Nota _____

Leia atentamente estas INSTRUÇÕES:

- ✓ Esta atividade contém 03 questões práticas. Verifique se ela está completa.
- ✓ Leia toda a prova antes de iniciar.
- ✓ Ao final da atividade salve a pasta scr (ou os arquivos de fonte) em um arquivo zipado com seu nome e contendo todas as atividades.
- ✓ Coloque seu nome no arquivo zipado.
- ✓ Salve o arquivo no seu pendrive como forma de backup.
- ✓ As questões são dependentes umas das outras. Caso não consiga implementar uma indique uma forma alternativa continuar as demais questões (comente no código).

Você recebeu a tarefa de preparar a hierarquia de classes abaixo para sua equipe de programação.

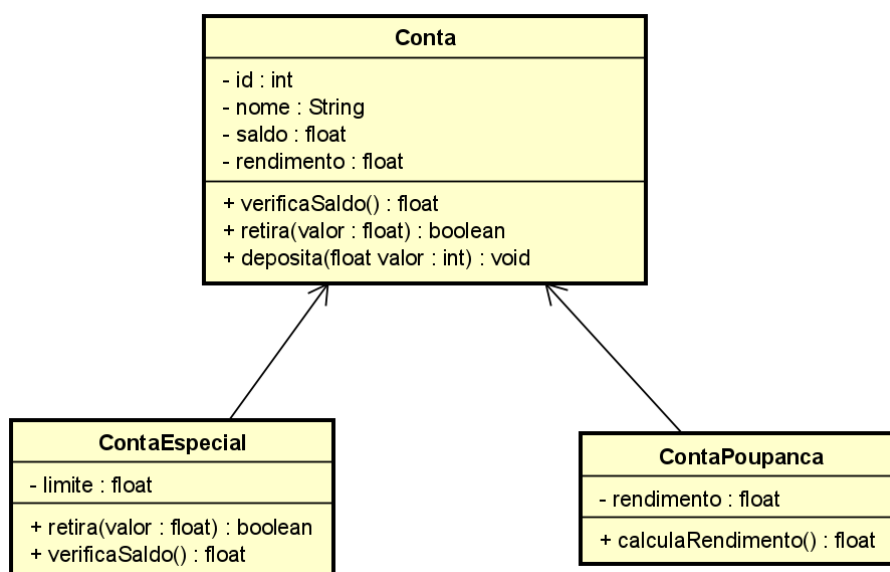


Figura 1 – Diagrama de classes Conta.

1) Implemente a hierarquia de classes conforme indicado na figura.

- ✓ Todas as classes **podem** ter construtores e funções get e set necessárias.
- a) Implemente a classe Conta.
- b) Implemente a classe ContaEspecial.
- c) Implemente a classe ContaPoupança.
- d) Todas as classes **devem** implementar a função toString que apresenta seus atributos (Sugestão: mesma saída do csv "id,nome,saldo,limite,rendimento").

Funções de Conta:

- e) verificaSaldo - retorna o saldo.
- f) retira - Verifica o saldo e, caso (valor <= saldo) remove o valor informado e retorna true, senão, retorna false.
- g) deposita - adiciona ao saldo o valor informado.

Funções de ContaEspecial:

- h) retira - Sobrecarrega as funções de Conta agora considerando saldo + limite.
- i) verificaSaldo - Sobrecarrega as funções de Conta agora considerando saldo + limite.

Funções de ContaEspecial:

- j) calculaRendimento - retorna o saldo * rendimento/100

- 2) Crie a classe GerenciadorDeContas que lê o arquivo contas.csv e armazena as contas no programa.
- ✓ Cada linha deve ter seu tipo de dado identificado (Conta-ContaEspecial-ContaPoupanca).
 - ✓ Organize os dados na classe gerenciadora conforme seu tipo.

Figura 2

| id | nome | saldo | limite | rendimento |
|----|-----------------|--------|--------|------------|
| 1 | Damaris Peggrem | -64.88 | | |
| 2 | Kaia Abotson | 690.64 | | 0.3 |
| 3 | Aluin Traut | 633.21 | 609 | |
| 4 | Novelia Beggs | 270.28 | 834 | |
| 5 | Clayton Bentley | 220.00 | | 0.47 |

Figura 2 - Formato do arquivo de texto e representação dos dados nesse arquivo.

Observações:

- A classe deve ter uma função do tipo "void carregaDados()" para ler e buscar os dados no arquivo.
- Esses dados devem ser armazenados em coleções (arrays ou arrays avançados).
- A função toString da classe fornece todos os dados no formato do arquivo de texto.
"id,nome,saldo,limite,rendimento"

3) Funções de verificação:

- Crie uma função principal main que deverá criar um objeto da classe GerenciadorDeContas para ser usada nas funções seguintes.
- Crie uma função que mostra na tela a quantidade de cada um dos tipos de conta.

Exemplo de saída:

| |
|--------------------|
| 350 Conta |
| 500 ContaEspecial |
| 200 ContaPoupanca. |

- c. Crie uma função que apresenta na tela uma lista de todas as contas em que o saldo é menor do que o limite. Por exemplo as contas de ID 1 e 15.

Exemplo de saída:

| |
|-------------------------------|
| id,nome,saldo,limite |
| 1,Damaris Peggrem,-64.88, 0 |
| 5,Clarey Bentley,-229.98, 0 |
| 14,Lanny Hawkyens,-942.09,206 |

- d. Crie uma função que apresenta na tela uma lista de todas as contas com seu respectivo calculo de rendimento.

Exemplo de saída:

| id,nome,saldo,calculoRendimento |
|---------------------------------|
| 2,Kaia Abotson,690.64, 0.69 |
| 5,Clarey Bentley,-229.98, 0 |
| 6,Nealson Babe.324.61. 3.21 |