



MAPA – Material de Avaliação Prática da Aprendizagem

Acadêmico: Yasser Ibrahim Abdallah Vaz Condoluci	R.A.: 222212725
Curso: Engenharia de software	
Disciplina: Paradigmas de linguagem de programação	
Valor da atividade: 3,00	Prazo: 16/09/2022

Instruções para Realização da Atividade

1. Todos os campos acima deverão ser devidamente preenchidos;
2. É obrigatória a utilização deste formulário para a realização do MAPA;
3. Esta é uma atividade INDIVIDUAL. Caso identificado cópia de colegas, o trabalho de ambos sofrerá decréscimo de nota;
4. Utilizando este formulário, realize sua atividade, salve em seu computador, renomeie e envie em forma de anexo;
5. Formatação exigida para esta atividade: documento Word, Fonte Arial ou Times New Roman tamanho 12, Espaçamento entre linhas 1,5, texto justificado;
6. Ao utilizar quaisquer materiais de pesquisa referencie conforme as normas da ABNT;
7. Critérios de avaliação: Utilização do Template; Atendimento ao Tema; Constituição dos argumentos e organização das Ideias; Correção Gramatical e atendimento às normas ABNT;
8. Procure argumentar de forma clara e objetiva, de acordo com o conteúdo da disciplina.

Em caso de dúvidas, entre em contato com seu Professor Mediador.

Bons estudos!



Os modelos utilizados neste MAPA são baseados na linguagem Java e em seu paradigma orientado a objeto.

Para as contas, será criada uma classe principal chamada “Conta” que herda os atributos de cliente (outra classe que será descrita logo adiante) onde terá os atributos:

- numeroAgencia: será do tipo Inteiro, por se tratar de um identificador da agência, foi utilizado o tipo inteiro pois se trata de um dado numérico inteiro.
- NumeroConta: receberá o tipo Inteiro, por se tratar de um identificador, assim como o número da agência, será utilizado o tipo de dado inteiro por ser um dado numérico sem ponto flutuante.
- Saldo: este receberá o tipo Real, pois trata-se de um valor numérico com ponto flutuante, números após a vírgula, neste dado o valor a ser armazenado é o dinheiro (saldo) e pode haver centavos.

Também será criado, para esta classe, métodos:

- Método depositar: este método fará uma ação que receber um valor a qual o responsável da conta informará para acrescentar ao saldo da conta-corrente ou conta poupança do mesmo.
- Método sacar: este método fará uma ação onde o responsável da conta informa um valor à qual conste no saldo sendo assim retirado da conta-corrente ou poupança.
- Método transferir: este método fará a transferência entre contas, neste método deverá ser informado o número da agência e da conta a qual deseja fazer a transferência.
- Método alterarSenha: por meio do polimorfismo será feita a sobrescrita do método onde validará a senha antiga ou algum outro dado pertinente e receberá uma nova senha informada pelo cliente para que consiga alterar de acordo com a conta sendo ela corrente ou poupança.

Após a criação da classe principal de conta, será criada uma outra classe chamada de “ContaCorrente” que herdará os atributos e métodos da classe principal e serão adicionadas novas funcionalidades a esta classe.

- `limiteChequeEspecial`: este atributo receberá o tipo `Real`, pois é um valor numérico com ponto flutuante, o dado que é armazenado neste atributo é um limite em dinheiro a qual existe a possibilidade de ter números após a vírgula.

Para a conta poupança, também será criado uma classe herdeira da classe principal “Conta” e também receberá atributos e métodos específicos para esta classe.

Para ela, será criado um atributo “`taxaJuros`” que é do tipo `real`, que se trata de um valor numérico com números após a vírgula, e terá um método chamado “`calcularJuros`” que receberá o valor depositado no método da classe principal “`depositar`” e será feito um cálculo de juros sobre este valor.

Para informações de clientes, será criada uma classe com o nome “`Cliente`” onde será atribuído as informações pessoais de cada cliente respectivamente:

- Atributo `nome`: recebe o tipo `String`, a qual é usado para armazenar uma cadeia de caracteres.
- Atributo `cpf`: Também receberá o tipo `String`, pois apesar de ser números, o CPF pode iniciar com 0 (zero), se utilizarmos um tipo inteiro por exemplo, os zeros que estão à esquerda serão ignorados fazendo que o CPF seja inválido.
- Atributo `dataNascimento`: Também receberá o tipo `String`, será usado esse tipo pois assim como o `cpf`, algumas datas iniciam com 0 zero e usando o tipo inteiro ele ignoraria o número zero.
- Atributo `endereço`: receberá o tipo `String`, por ser um texto.
- Atributo `telefone`: tipo `String`, assim como o `cpf` e a data um número de telefone pode iniciar com o número zero sendo assim ignorado caso utilize o tipo inteiro.
- Atributo `senha`: tipo `String`, sendo usado esse tipo por armazenar texto.

Será criado um método chamado “`alterarSenha`”.

- Método `alterarSenha`: Este método validará a senha antiga ou algum outro dado pertinente e receberá uma nova senha informada pelo cliente para que consiga alterá-la.