

Solução ao Pedido 7:

Os códigos das classes do pedido número 7 podem ser baixados do moodle, analisados e comparados com o que foi feito pelo aluno. Doravante utilizaremos como padrão essas classes.

Pedido 8:

*Elabore a documentação dos códigos a serem gerados.*

Implemente a classe Account sabendo que:

Representa uma conta bancária com quatro atributos.

A classe tem os seguintes métodos:

Construtor com quatro argumentos que inicializa os atributos;

private boolean validatePIN(int userPIN);

public double getAvailableBalance() que retorna o saldo disponível na conta;

public double getTotalBalance() que retorna o saldo total da conta;

public void credit() que credita uma quantia na conta especificada;

public void debit() que debita uma quantia na conta especificada;

public int getAccountNumber() que retorna o número da conta;

Tempo para realizar a tarefa: 15 minutos.

Pedido 9:

*Elabore a documentação dos códigos a serem gerados.*

Implemente a classe BankDatabase sabendo que:

Por ora o banco de dados será modelado como um array com as informações de conta dos usuários. Por simplicidade serão construídas, apenas, duas contas;

Essa classe tem zero ou mais objetos da classe Account, por força de um relacionamento de composição com aquela classe.

A classe tem os seguintes métodos:

Construtor sem argumentos que inicializa as contas;

Private Account getAccount(accountNumber) que recupera o objeto que contém o número da conta especificado como parâmetro;

public boolean authenticateUser(int userAccountNumber, int userPIN) que determina se o número da conta e PIN correspondem a uma conta do banco;

public double getAvailableBalance(int userAccountNumber) que retorna o saldo disponível na conta;

public double getTotalBalance(int userAccountNumber) que retorna o saldo total da conta;

public void credit(int userAccountNumber, double amount) que credita uma quantia na conta especificada;

public void debit(int userAccountNumber, double amount) que debita uma quantia na conta especificada;

Observações: Essa classe funciona como um intermediário entre a classe ATM e os objetos Account reais. Assim, os métodos dessa classe nada mais fazem do que invocar métodos correspondentes do objeto Account pertencente ao atual usuário do ATM.

Tempo para realizar a tarefa: 20 minutos.