

# Trabalho de Programação Orientada a Objetos I

Elaine Faria  
elaine@ufu.br

Programação Orientada a Objetos 1 (Semestre 02/2019)  
Bacharelado em Sistemas de Informação  
Faculdade de Computação  
Universidade Federal de Uberlândia

## 1. Introdução

O objetivo principal do trabalho é modelar e implementar um sistema de acordo com os requisitos descritos a seguir usando as melhores práticas da orientação a objetos e a linguagem de programação Java.

- Cada grupo deverá ser composto por no máximo 3 alunos.
- Valor: 30 pontos.
- Data da Entrega do material: 02/12
- Data da Apresentação Final: 04/12, 09/12 e 11/12/19 (dependendo do número de grupos serão necessários mais dias de apresentação)

## 2. Descrição do Trabalho

Os alunos deverão modelar, usando o diagrama de classes da UML, e implementar um sistema a partir do conjunto de requisitos descritos na seção 3. Além disso, cada grupo deverá definir 2 requisitos adicionais ao problema, descrevê-los, modelá-los e implementá-los. O sistema desenvolvido deverá utilizar recursos de interface gráfica (pacote Swing ou Awt do Java).

Os requisitos adicionais definidos pelos alunos devem gerar uma nova classe, ou um novo atributo (desde que não seja de um tipo primitivo) ou um novo método.

Quando os requisitos descritos não forem suficientes para fazer a modelagem, os alunos devem descrever o que eles estão supondo como requisitos para fazer a modelagem. Ex: se em um dado requisito gerar uma associação entre duas classes, mas não for possível estabelecer sua multiplicidade, os alunos deverão descrever o que será assumido nesse caso.

O trabalho consiste das seguintes tarefas:

Tarefa 1: Modelar o sistema descrito a seguir, identificando suas classes, atributos, métodos, etc. Para esta atividade, o diagrama de classes da UML deve ser usado. É importante que os requisitos adicionais definidos pelos alunos sejam também descritos, modelados e implementados. Para cada um dos itens descritos na seção 3, os alunos devem criar um documento informando como estes foram modelados no diagrama de classes. O não cumprimento deste item gerará em perda de 50% da nota final do trabalho.

Tarefa 2: Implementar o sistema que foi modelado na tarefa 1 usando a linguagem Java. É muito importante que o código esteja documentado e que as boas práticas da orientação a objetos tenham sido usadas. Nesta tarefa será avaliado o uso correto de cada um dos conceitos vistos em sala de aula sobre programação orientada a objetos. Todas as informações do sistema poderão ser armazenadas na memória. Como não é foco dessa disciplina trabalhar com Banco de Dados, todos os dados ficarão na memória durante a execução do programa. Ao encerrar o sistema, as informações cadastrais deverão ser armazenadas em arquivos. O arquivo deverá ser gravado somente quando o usuário encerra o programa. Use um arquivo texto para cada classe. Neste momento, o arquivo texto pode ser sobrescrito.

Tarefa 3: Criar uma interface gráfica que facilite a interação do usuário como sistema desenvolvido na tarefa 2. Para isso, recursos do pacote Swing ou Awt do Java devem ser usados.

**Data Entrega: 02/12/19**

### 3. Requisitos do Sistema

#### **Sistema para uma Instituição Financeira**

- a) Para a instituição financeira é muito importante manter um cadastro de todas as contas que foram criadas na instituição. No momento de criação da conta é importante associar uma senha a ela. Essa senha será solicitada antes da execução de qualquer transação.
- b) Para cada conta criada é importante saber se a conta está ativa ou já foi desativada (ou seja, o cliente encerrou a conta). Isso ajudará a filtrar as pesquisas. Ainda que um cliente encerre uma conta no banco, é importante manter o registro da conta na base de dados, mas com um indicativo de que ela não está ativa.
- c) As contas da instituição podem pertencer às seguintes categorias: corrente, poupança e salário.
- d) Para as contas da categoria corrente, as seguintes informações devem ser armazenadas: nro da conta, saldo atual, data de abertura, data da última movimentação, limite do cheque especial e valor da taxa administrativa.
- e) Para as contas da categoria poupança as seguintes informações devem ser armazenadas: nro da conta, saldo atual, data de abertura, data da última movimentação, rendimento do mês atual.

- f) Para as contas da categoria salário as seguintes informações devem ser armazenadas: nro da conta, saldo atual, data de abertura, data da última movimentação, limite para saque e limite para transferência.
- g) Os clientes da instituição devem ser registrados com as seguintes informações: CPF, nome, endereço completo, estado civil, escolaridade, data de nascimento.
- h) É importante também saber em qual agência o cliente foi cadastrado.
- i) As agências bancárias devem ser previamente cadastradas, sendo que cada agência possui um número, um nome fictício e um endereço, sendo que os campos cidade, estado e bairro são campos frequentemente usados para se realizar buscas por agências.
- j) Cada agência possui um gerente, que é um funcionário nomeado para tal tarefa. Cada gerente pode gerenciar apenas uma agência.
- k) Os gerentes são funcionários da empresa que precisam ter armazenados alguns atributos específicos: data de ingresso na carreira de gerente, agência que ele gerencia e se possui curso de formação básico em gerência.
- l) Para os funcionários da agência é preciso armazenar as seguintes informações: CPF, nome completo, nro da carteira de trabalho, RG, data de nascimento, endereço, sexo, estado civil, cargo na empresa (existem diferentes cargos além do gerente) e salário.
- m) Toda conta bancária deve ter no mínimo um cliente associado. No entanto, uma conta pode ser conjunta. Nesse caso, é possível ter dois clientes associados à mesma conta. Um mesmo cliente pode ter diferentes contas bancárias.
- n) Toda conta bancária está ligada a uma agência bancária, ou seja, a agência na qual ela foi criada.
- o) Uma vez que um cliente tenha uma conta bancária ele pode efetuar diferentes movimentações financeiras, conhecidas como transações bancárias. Cada transação bancária está relacionada a uma conta bancária. Além disso, é importante armazenar a data em que a transação foi realizada, o valor da transação e o canal onde foi feita a transação (*internet banking*, caixa eletrônico ou caixa físico). Toda vez que uma transação é realizada o saldo do cliente deve ser atualizado.
- p) Cada transação possui um tipo. Os tipos de transação atualmente disponíveis são: saque, transferência, depósito e pagamento. Em cada uma dessas transações é importante armazenar apenas o valor da transação. Com o objetivo de simplificar o trabalho não vamos armazenar os atributos específicos de cada tipo de transação, embora esse seja um requisito interessante em problemas reais.

É importante ter um usuário administrador do sistema, responsável por manter agências e funcionários. Um funcionário gerente pode manter clientes e contas (o administrador também pode). O usuário cliente realiza transações bancárias e visualiza os dados da sua conta.

#### 4. Material a ser entregue

O código fonte, a modelagem UML, a descrição dos requisitos adicionais e a descrição de como foi implementado cada um dos requisitos deverão ser

entregues ao professor por email. A modelagem UML, a descrição dos requisitos adicionais e a descrição de como foi implementado cada um dos requisitos deverão ser entregues em versão pdf. Além do código fonte (arquivos .Java), um arquivo executável também deve ser enviado (.jar).

Email: [elaine@ufu.br](mailto:elaine@ufu.br)

É importante que o título do email seja: GSI015- Trabalho

No corpo do email deve aparecer o nome dos integrantes do grupo.

## 5. Apresentação

Cada apresentação terá duração de no máximo 20 minutos. Todos os integrantes do grupo deverão apresentar o trabalho. A apresentação consiste em responder perguntas feitas pelo professor, que são direcionadas a cada um dos integrantes do grupo. A nota final de cada aluno será ponderada pela nota da apresentação (nota da apresentação: entre 0 e 1).

## 6. Regras

- Não serão aceitos trabalhos atrasados. Se o grupo não entregar o trabalho no dia combinado, ele receberá nota zero.
- Atenção com o envio de email ao professor. Só serão considerados os emails recebidos pelo professor até às 24h do dia da entrega do trabalho. Trabalhos recebidos fora do prazo receberão nota zero.
- Em caso de projetos copiados de colegas todos os envolvidos recebem nota zero. Lembre-se é muito improvável que haja trabalhos totalmente iguais.
- O professor não ajudará os grupos na construção do trabalho.
- O professor poderá tirar dúvidas conceituais em horário de aula ou horário de atendimento.
- O professor poderá questionar cada um dos integrantes do grupo no momento da apresentação.
- A nota dos integrantes não necessariamente será a mesma. Se durante a apresentação o professor detectar que algum integrante do grupo não tem domínio sobre o projeto, ele poderá receber uma nota menor que os demais integrantes.