



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Disciplina: **Programação Orientada a Objetos 1**

Curso: **Sistemas de Informação**

Professora: **Elaine Ribeiro Faria**

Aula Prática no 14

Tema: *ArrayList* e Arquivos

Instruções:

- 1- Faça a codificação seguindo o enunciado usando a linguagem Java.
- 2- Crie um arquivo txt chamado IntegrantesGrupo.txt contendo:
Nome e nro de matrícula dos integrantes do grupo
- 3- Envie pelo Microsoft Teams um arquivo. Zip contendo o código fonte (arquivos. Java do seu projeto) e o arquivo .txt

Data da entrega: 15/12/20 às 23:59

Avaliativa: **SIM**

Perguntas:

1. Considerando sistema de Instituição Financeira desenvolvido nas aulas práticas 6, 8 e 12, você deverá complementar a implementação em Java de acordo com o que se pede a seguir:
 - a) Implemente a classe Conta como abstrata.
 - b) Implemente um método abstrato calculaTarifa() na classe Conta. Note que um método abstrato não possui implementação. Cada subclasse tem sua própria implementação. Isto é conveniente, pois o cálculo da tarifa segue uma regra diferente para cada tipo de conta.
 - c) Para a classe conta corrente, a tarifa é um valor fixo por mês, que depende da categoria da conta (*standard* ou *premium*). Para a classe poupança, a tarifa é um valor fixo, igual para todas as contas. Para a conta salário, a tarifa é 0,1% do saldo atual.
 - d) Transforme a superclasse (Pessoa) de Cliente e Funcionário em classe abstrata.
 - e) Crie classes especiais capazes de manter os dados gerados no sistema. Por manter, entende-se: cadastrar, consultar e excluir. Para isto, crie as classes “DadosClientes”, “DadosContas”, “DadosAgencias”, etc.. Estas classes serão responsáveis por encapsular o acesso a cada tipo de dado específico. Para isto, estas classes devem implementar um *ArrayList* privado para armazenar a informação, e métodos públicos para permitir o acesso à informação (inserir um novo objeto, buscar

um objeto, excluir um objeto, etc.). Os objetos de dados deverão ser criados apenas uma vez na classe principal e utilizados ao longo da execução da aplicação. **Exemplo:**

```
class DadosClientes{
    private ArrayList<Cliente> clientes = new ArrayList<Cliente>();
    public void cadastrar(Cliente c) {
        this.clientes.add(c); //ADICIONA O CLIENTE NO ARRAY
        System.out.println("Total de Clientes inseridos...: ");
        System.out.println(this.clientes.size());
    }
    public void listar(){
        for (Cliente objeto: this.clientes) {
            objeto.mostrarDados();
            //método mostrarDados();
        }
    }
    //este método retorna o objeto Cliente caso encontrado, ou null,
    caso não encontrado
    public Cliente buscar(String CPF) {
        Cliente c = null;
        for ( Cliente objeto: this.clientes) {
            if (objeto.getCPF().equals(CPF)) {
                c = objeto;
                break;
            }
        }
        return c;
    }
    //este método usa o método buscar já implementado
    public boolean excluir(String CPF){
        Cliente c = this.buscar(CPF);
        if (c != null) {
            this.clientes.remove(a);
            return true;
        }
        else {
            return false;
        }
    }
}
```

- f) Acrescente ao seu programa uma funcionalidade para salvar os *ArrayLists* de dados em um arquivo binário. Crie um arquivo para cada *ArrayList*.
- g) Invente um requisito no sistema que justificasse a criação de uma interface no sistema