

Trabalho prático Swing e JDBC.

Valor: 100% da quarta nota.

Equipes: Máximo 3 pessoas.

Uma empresa resolveu implantar um novo sistema de recursos humanos com autenticação em cada sistema. O novo sistema precisa ser em Java com interface SWING e seguindo o paradigma orientado a objetos. O sistema terá os seguintes requisitos:

1. Uma tela para manter (incluir, atualizar, excluir e listar) funcionários (Nome, sobre nome, RG, CPF, Telefone, cargo, nível, departamento, senha)
 - a. Nesta tela deve ser possível listar todos os Funcionários (exceto senha) (Use AbstractTableModel)
 - b. Deve ser possível atualizar os dados de um Funcionário, clicando no registro da tabela.
 - c. Deve ser possível excluir Funcionários
 - d. Deve ser possível listar os funcionários por nome (ou parte do nome), por sobrenome (ou parte), por RG, por CPF e por cargo.
 - e. Deve ser possível ordenar a lista por nome, sobrenome(ordem alfabética) ou salário (do maior para o menor). Implemente a interface Comparable.
2. Uma tela para manter (incluir, atualizar, excluir e listar) os sistemas da empresa (Nome do Sistema)
 - a. Fazer as devidas validações para atualizar e excluir um sistema.
3. Uma tela para cadastrar/manter os departamentos da empresa (Nome do Departamento)
 - a. Fazer as devidas validações para atualizar e excluir um departamento.
4. Uma tela para atualizar o salário por cargo e nível
5. Os seguintes cargos e níveis devem existir no sistema. Crie uma classe para cada cargo herdando de funcionário:
 - a. Diretor (Nível 1, 2 e 3)
 - b. Gerente (Nível 1,2 e 3)
 - c. Analista (Nível 1,2 e 3)
 - d. Programador (Nível 1,2 e 3)
6. Um diretor pode gerir mais de um departamento. Quando o cargo do Funcionario for Diretor, um novo campo aparecerá na tela de cadastro (Departamento Gerenciado). Este campo será uma lista de departamentos, onde o usuário escolherá o(s) departamento(s).
7. Um gerente é responsável por apenas um departamento. Quando o cargo do Funcionario for Gerente, um novo campo aparecerá na tela de cadastro (Departamento Gerenciado). Este campo será uma comboBox, onde o usuário escolherá o departamento.
8. Uma tela para imprimir um relatório de Funcionários com seus respectivos salários e bônus anual. Utilize polimorfismo, ou seja, uma lista de Funcionários.
9. O cálculo do bônus anual segue as seguintes regras. Implemente estas regras utilizando conceitos de polimorfismo.
 - a. Para Diretor:
 - i. 4 salários + R\$3000,00 por departamento gerenciado
 - b. Para Gerente
 - i. 2 salários + R\$100,00 por pessoa gerenciada
 - c. Analista
 - i. 1 salário
 - d. Programador
 - i. 0,8 salário
 - e. Auxiliar de Limpeza
 - i. 0,5 salário
10. Uma tela para cadastrar a autorização para os sistemas de um funcionário
 - a. Nesta tela deve ser possível encontrar um funcionário por CPF.
 - b. Relacionar o funcionário a um ou mais sistemas.
11. Implementar uma interface Autenticavel. Esta interface possui um método com a seguinte assinatura: boolean autentica(String nomeSistema, String usuário, String senha); O Funcionário deve implementar esta interface. A implementação deve comparar usuário com CPF, senha com senha do funcionário e verificar se o nomeSistema passado como parâmetro faz parte da lista de sistemas do Funcionario.
 - a. O Auxiliar de Limpeza não pode ter acesso a nenhum sistema, mesmo que se cadastre na tela de autorizações. O método autentica deve ser sobrescrito para o Auxiliar de Limpeza, sempre retornando false.
 - b. O Diretor pode ter acesso a todos os sistemas, mas o sistema deve validar usuário e senha.
12. Uma tela para login de Funcionário. O Funcionário informa o usuário, senha e escolhe em uma ComboBox o sistema que deseja fazer login. Se o login para aquele sistema for válido, uma mensagem de sucesso será mostrada: "Você está logado no sistema X.". Se o login para aquele sistema for inválido, outra mensagem deve ser retornada: "Você não possui permissão para acessar o sistema X ou seu login ou senha estão inválidos.". Não é necessário utilizar criptografia para senhas.

O programa acima deve ser feito utilizando as seguintes tecnologias:

- Utilizar herança para definir as classes Pessoa, Funcionario, Diretor, Gerente, Analista, Programador, Auxiliar de Limpeza. Utilize classes abstratas quando necessário.
- Utilizar polimorfismo no cálculo do bônus de cada funcionário.
- Utilizar Interface para implementar a autenticação. Utilizar polimorfismo na autenticação.
- Java Swing
- JDBC para persistência em banco de dados. Utilize o Design Pattern "Data Access Object" para persistência.

Itens para serem entregues:

1. Diagrama de classes
2. Diagrama de E-R
3. Projeto na IDE Eclipse ou Netbeans com código fonte
4. Script para criação do banco de dados
5. Arquivo .jar executável

Avaliação:

- Material entregue
- Qualidade do software (bugs encontrados na defesa)
- Defesa do código e a nota será individual, considerando a defesa