Atividade 1: Modelagem Orientada a Objetos

POO - BSI - Ifes Serra

3 de abril de 2024

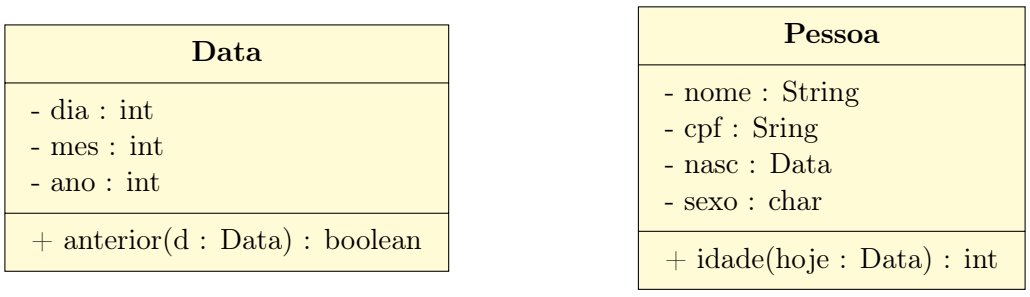
1 Diagrama de Classes UML

A Linguagem de modelagem unificada (UML) ajuda a modelar sistemas de diversas maneiras. Um dos tipos mais populares na UML é o **diagrama de classes**, que descreve o que deve estar presente no sistema a ser modelado incluindo as classes, seus atributos, suas operações e seus relacionamentos com outras classes. A UML foi criada para padronizar a descrição dos diagramas OO. Um diagrama de classes permite ilustrar modelos de dados para sistemas de informação (simples ou complexos), entender melhor a visão geral dos esquemas de uma aplicação, expressar visualmente as necessidades específicas de um sistema, e fornecer uma descrição do sistema independente de implementação.

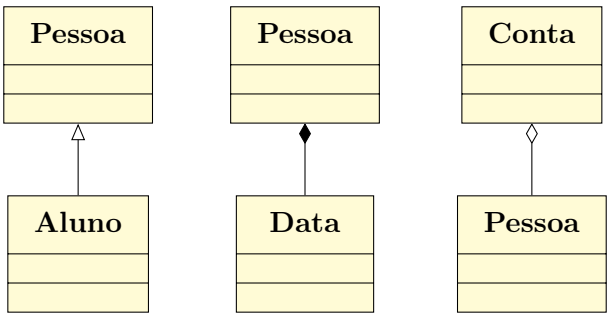
Uma classe no diagrama é representada por um retângulo dividido em três partes:

- A parte superior contém o nome da classe e é sempre necessária.
- A parte do meio contém os atributos da classe. Cada atributo é exibido em uma linha separada. Cada linha possui o nome do atributo e seu tipo, separados por um ":".
- A parte inferior inclui as operações da classe (métodos). Cada operação é exibida em uma linha separada. O método é descrito por seu nome e sua assinatura (lista de parâmetros entre parênteses e separados por vírgula, e o tipo do retorno do método após o ":").

Atributos e métodos possuem modificadores de acesso, que definem sua visibilidade: público (+), privado (-) e protegido (#). Métodos e atributos estáticos são representados sublinhados. Exemplo:



Diagramas de classe também podem exibir o relacionamento entre as classes do sistema. Nesta atividade, iremos considerar os seguintes tipos de relacionamentos:

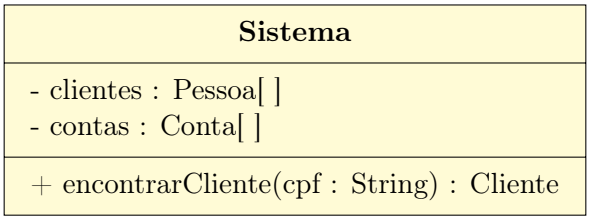


1. A seta entre as classes Aluno e Pessoa indica **herança** entre elas. Neste caso, Aluno herda de Pessoa. Um objeto da classe Aluno também "é um" objeto da classe Pessoa, mas pode conter atributos e métodos próprios.

2. O losango fechado entre Pessoa e Data indica uma relação de **composição**. Neste caso, podemos dizer que a classe Pessoa "tem" um objeto da classe Data (a data em que a pessoa nasceu). Por ser uma composição, o objeto da classe **Data** vai ser instanciado no momento em que a avaliação for criada e existirá apenas ali dentro do objeto da classe **Pessoa**. Neste exemplo, o sistema não precisa armazenar uma Data sem que ela esteja associada ao uma Pessoa específica. Quando o objeto da classe Pessoa for destruído, o objeto com sua data de nascimento será destruído junto. Podemos dizer que a data é uma "parte da" Pessoa.

3. O losango aberto entre Conta e Pessoa indica uma relação de **agregação**. Novamente, podemos dizer que a classe Conta "tem" um objeto da classe Pessoa (o titular da conta). Entretanto, neste caso, o objeto Pessoa não será instanciado apenas quando a Conta for criada. A Pessoa pode existir no sistema sem a Conta existir. Quando a conta é criada, as pessoas já existiam no sistema, e a conta faz apenas uma referência a uma delas como sendo o titular dessa conta. Quando o objeto da classe Conta for destruído, seu titular continuará existindo no sistema (ele pode ter ou vir a ter outras contas no banco, por exemplo).

Por fim, neste trabalho, usaremos uma notação única para representar um conjunto de objetos (independente de este conjunto ser armazenado como um array, uma lista, etc). No exemplo abaixo, por exemplo, a classe sistema possui um conjunto de pessoas e um conjunto de contas:



2 O Problema: Aplicativo de Fotos

Você é viciado em um aplicativo em que posta suas fotos na internet. Você usa o aplicativo não apenas para se exibir para os demais usuários, mas também para buscar possíveis interesses amorosos. Porém, você acha que o aplicativo não está facilitado essa sua busca por um amor, e sente falta de algumas funcionalidades que não encontra no sistema. Por exemplo, quando você viaja, sente vontade de filtrar todos os usuário que você segue que morem naquela cidade. Além disso, você acha que seguir um usuário pode não demonstrar o real interesse que você tem nele. Além de curtir as fotos de um certo usuário, você gostaria de curtir *o próprio usuário* e saber quando esse interesse for mútuo.

3 As Classes e Atributos

Você, portanto, decidiu implementar seu próprio aplicativo de fotos. Porém, para ganhar dinheiro, você vai diferenciar os usuários entre pessoas e empresas (os dois tipos podem postar fotos igualmente).

Os dois tipos de usuários possuem login, nome, senha, um local (cidade/Estado/país), um conjunto de postagens, um conjunto de usuários que ele segue, um conjunto de seguidores e um conjunto de pessoas interessadas nele. Se o usuário for uma empresa, entende-se que a pessoa tem interesse em seus produtos, e não um interesse amoroso. Os locais não devem ser um texto qualquer (existem cidades, Estados e países previamente cadastrados, e o usuário deve selecionar um deles).

Uma pessoa possui ainda cpf, sexo, data de nascimento, biografia (um pequeno texto no qual ela se descreve) e um conjunto de usuários nos quais ela possui interesse.

Já uma empresa possui cnpj, endereço, site, descrição e área de atuação. A área de atuação não deve ser um texto qualquer (existem áreas de atuações previamente cadastradas, e a empresa deve selecionar uma delas).

Cada postagem feita por um usuário é composta por uma foto, uma legenda e uma data de postagem. Pode-se considerar que a foto é apenas uma String com sua localização no banco de dados de fotos (ex: "diaDosPais.jpg").

Por fim, o sistema deve armazenar um conjunto com todas as pessoas cadastradas e outro conjunto com as empresas.

4 As funcionalidades

Uma data deve possuir um método que compare-a com uma outra data recebida como parâmetro. Este método vai retornar um número inteiro (-1, se a data em questão for anterior à outra, 0 caso sejam iguais e 1 caso seja posterior). Além disso, deve possuir um método para exibi-la no formato "20/12/2024".

Um usuário possui métodos para validar seu acesso (dada sua senha), alterar sua senha (dada a senha atual e a nova), postar uma foto (dada a foto, a legenda, a data de hoje e a senha do usuário), filtrar postagens que ocorreram entre duas datas (dadas estas duas datas e retornando um conjunto de postagens) e seguir um outro usuário (dado o usuário a ser seguido).

Uma pessoa, especificamente, possui métodos para validar seu acesso (aqui, além da senha, recebendo seu CPF e data de nascimento para melhorar a validação), alterar senha (com os parâmetros extras necessários para validar a senha de uma pessoa), calcular sua idade (dada a data de hoje), curtir uma postagem (dada a postagem), curtir um usuário (dado o usuário), e listar seus matches (pessoas que ela curtiu e que curtiram ela também).

Já uma empresa possui métodos para validar seu acesso (aqui, além da senha, recebendo seu CNPJ para melhorar a validação), alterar senha (com os parâmetros extras necessários para validar a senha de uma empresa), promover uma postagem (dada a postagem, o tempo em dias para a postagem ser promovida e a data de hoje) e enviar uma mensagem a todos os usuários que seguem ou curtiram a empresa (dado o texto com a mensagem).

Uma postagem possui um método para editar sua legenda (dada a nova legenda). Não é possível alterar a foto já postada. Uma vez na internet, sempre na internet.

Por fim, o sistema possui métodos para buscar usuários (dado o local do usuário e o conjunto em que ele será buscado - este método será estático e retorna um conjunto de usuários), buscar um conjunto de pessoas pelo nome (dado seu nome), buscar um conjunto de pessoas dentro de uma faixa etária (dada uma idade mínima, uma idade máxima e a data de hoje), buscar um conjunto de pessoas pelo local onde moram (dado o local), buscar um conjunto de empresas de

uma certa área de atuação (dada a área) e buscar um conjunto de empresas pelo local (dado o local).

5 O diagrama de classes UML

Nesta primeira atividade, você deve criar ***apenas*** o diagrama de classes UML do sistema descrito neste documento, contendo todas as classes, atributos, métodos e relacionamentos entre as classes.

É importante que você utilize corretamente, sempre que pertinente, as boas práticas de orientação a objetos, como **herança**, **reescrita de métodos**, **sobrecarga de métodos**, **encapsulamento**, reuso de código, etc.

Não é necessário incluir os construtores das classes no diagrama, nem se preocupar com a leitura dos dados.

Você pode utilizar programas específicos para modelagem OO, editores de imagens ou até mesmo desenhá-lo num papel. Deve ser enviado apenas um arquivo único contendo o diagrama completo (pdf, jpeg ou png).

6 Observações

1. Esta atividade vale 20 pontos e deve ser entregue até 09/04.
2. Deve ser enviado apenas um PDF único ou um arquivo único com a imagem do diagrama de classes UML de sistema, considerando as boas práticas de POO (como reuso de código, por exemplo, sempre que possível).
3. A atividade pode ser feita em grupos de até **dois** integrantes.
4. Atividades entregues após o prazo serão automaticamente rejeitadas.
5. Atividades consideradas plágio terão nota 0 para quem copiou e para quem forneceu a atividade, e serão enviadas ao Conselho de Ética.
6. A atividade deve ser enviada na sala da disciplina do AVA.
7. Em caso de dúvidas na especificação da atividade ou na própria atividade, contate-me em sala de aula.