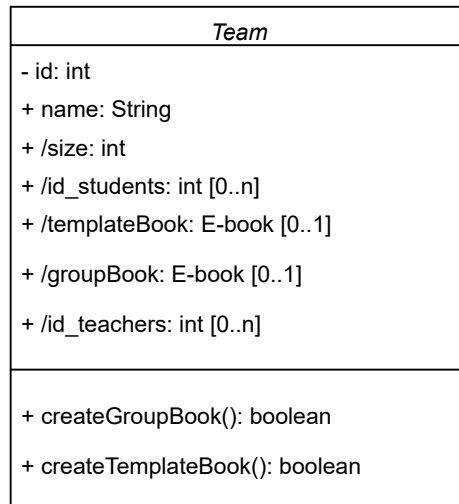


A) Defina as classes (em UML) para este cenário considerando os elementos: objetos, atributos e métodos. Não se preocupe com o relacionamento entre classes.

R: Serão apresentadas as classes em UML e logo a baixo uma breve descrição de seus atributos e métodos.

UML da classe Team



Descrição dos Atributos

id: identificação numérica da turma.

name: nome da turma.

size: quantidade de alunos da sala.

id_students: variável que armazena um array contendo o id de cada estudante da turma.

templateBook: variável que armazena o E-book da turma.

groupBook: armazena o groupNook da turma.

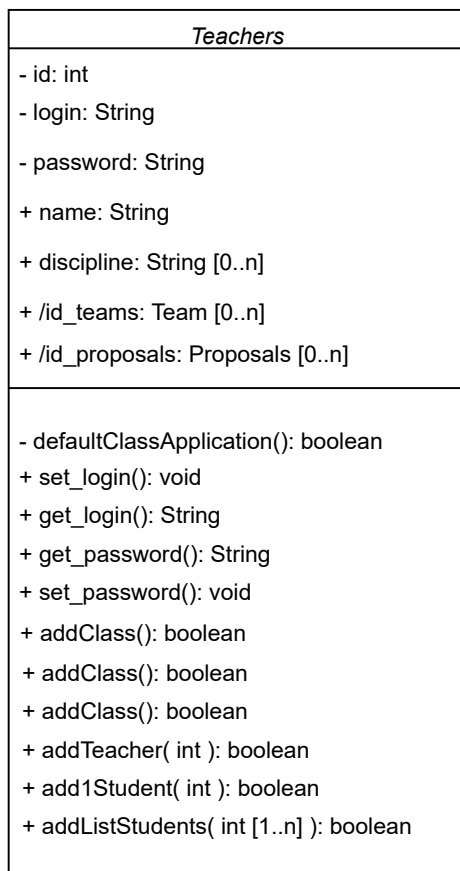
id_teachers: armazena num array o id de um a n professores da turma.

Descrição dos Métodos

createTemplateBook(): cria um livro para servir de orientação ao trabalhos dos alunos da turma.

creatGroupBook(): cria o livro colaborativo da turma.

UML da classe Teachers



Descrição dos Atributos

id: identificação numérica do professor.

name: nome do professor.

login: string que possibilita o acesso ao sistema.

password: senha de acesso.

discipline: vetor contendo as disciplinas que o professor leciona.

id_teams: guarda as identificações das turmas que pertencem ao professor.

id_proposals: guarda os planos de aula que pertencem ao professor.

Descrição dos Métodos

defaultClassApplication(): método que irá ser chamado pela função que cria uma turma, serve para o professor definir um usuário e senha padrão para a turma.

set_login: seta o atributo login

get_login: retorna o valor do atributo login

set_password: seta o atributo senha

get_password: retorna o valor do atributo senha

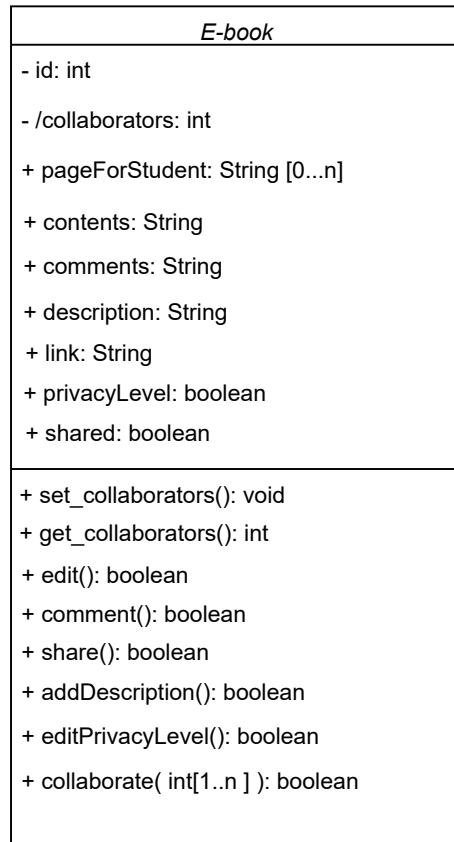
addListStudents(int [1..n]): adiciona uma lista de estudantes, passando um array contendo o id de cada um como parâmetro.

addClass(): Cria uma turma.

addTeacher(int): adiciona um professor a turma, passando como parâmetro o id do mesmo,

add1Student(int): adiciona um estudante a turma, passando como parâmetro o id do mesmo,

UML da classe E-book



Descrição dos Atributos

id: identificação numérica do E-book

collaborators: quantidade de colaboradores do e-book.

pageForStudent: como cada estudante tem sua página, essa variável armazena o número de cada uma.

contents: onde ficará armazenado o conteúdo dos e-books.

comments: String de comentário.

description: String com a descrição do e-book.

link: String com o link do e-book.

privacyLevel: variável que indica se é público (1) ou privado (0).

shared: indica se o e-book está sendo compartilhado (1) ou não está sendo compartilhado (0).

edit(): indica se o e-book está sendo editado (1) ou não (0).

Descrição dos Métodos

set_collaborators: seta o atributo collaborators.

get_collaborators: retorna o valor do atributo collaborators.

comment(): método que permite comentar no e-book e retorna um valor booleano indicando se o e-book foi comentado.

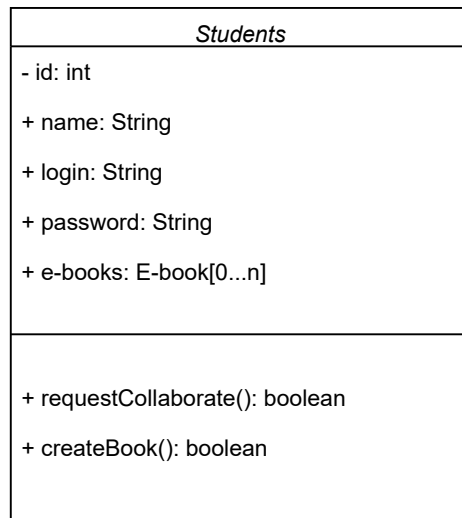
share(): método de compartilhamento do e-book e indica se está sendo compartilhado ou não.

addDescription(): método que adiciona uma descrição ao e-book retornando se há descrição.

editPrivacyLevel(): método que permite alterar o e-book entre privado ou público.

collaborate(int[1..n]): método que adiciona um colaborador ao e-book, passando como parâmetro o id do mesmo.

UML da classe Students



Descrição dos Atributos

id: número de identificação do estudante.

name: nome do estudante.

login: string de acesso ao sistema.

password: senha de acesso ao sistema.

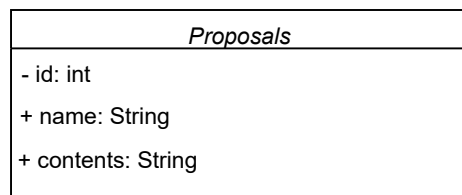
e-books: guarda os e-books criados pelo estudante.

Descrição dos Métodos

createBook(): método de criação de e-book, retorna se está sendo criado ou não.

requestCollaborate(): método que faz a solicitação para professor para ser um colaborador, caso o GroupBook tenha sido criado previamente pelo professor o aluno já entra direto, e por fim, o método retorna se está sendo solicitado ou não.

UML da classe Proposals



Descrição dos Atributos

id: número da identificação das propostas (planos de aula).

name: string com o nome da proposta.

content: conteúdos das propostas

OBSERVAÇÃO : esta classe não possui métodos não triviais.

B) Em sua visão, quais objetos podem ser instanciados das classes definidas? Justifique a sua resposta.

R: Os objetos que podem ser instanciados a partir das classes definidas são: *team* (turma), *teachers* (professor), e-book, *students* (estudante) e *proposals* (proposta). Tendo em conta que, objetos de mesma classe sempre serão distintos.