# Atividade de revisão - MATA55 PROGRAMAÇÃO ORIENTADA A OBJETOS - Turma 010100 Bruno de Sousa Behrmann Reginaldo Silva de Oliveira

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

Team
- id: int
+ name: String
+ /size: int
+ /id_students: int [0n]
+ /templateBook: E-book [01]
+ /groupBook: E-book [01]
+ /id_teachers: int [0n]
t and ata Charles Back () a had also
+ createGroupBook(): boolean
+ createTemplateBook(): boolean

#### 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 teachears: 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**

### Teachers

- id: int

- login: String

- password: String

+ name: String

+ discipline: String [0..n]

+ /id teams: Team [0..n]

+ /id\_proposals: Proposals [0..n]

- defaultClassApplication(): boolean

+ set\_login(): void

+ get login(): String

+ get\_password(): String

+ set\_password(): void

+ addClass(): boolean

+ addClass(): boolean

+ addClass(): boolean

+ addTeacher( int ): boolean

+ add1Student( int ): boolean

+ addListStudents( int [1..n] ): boolean

#### 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 atritbuto login

set\_password: seta o atributo senha get\_password: retorna o valor do atritbuto 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,

#### E-book

- id: int

- /collaborators: int

+ pageForStudent: String [0...n]

+ contents: String+ comments: String+ description: String

+ link: String

+ privacyLevel: boolean + shared: boolean

+ set\_collaborators(): void + get\_collaborators(): int

+ edit(): boolean

+ comment(): boolean + share(): boolean

+ addDescription(): boolean+ editPrivacyLevel(): boolean

+ collaborate( int[1..n ] ): boolean

### 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**

Students
- id: int
+ name: String
+ login: String
+ password: String
+ e-books: E-book[0n]
+ requestCollaborate(): boolean
+ createBook(): boolean

# 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**

Proposals
- id: int
+ name: String
+ contents: String

#### 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.