



**Disciplina:** Projeto de Programação  
**Curso:** Ciência da Computação/Sistemas de Informação  
**Docente:** Profª Ma. Layse Souza  
**Ano:** 2024.2

**Nome dos Alunos:**  
**Antônio Fernandes De Santana Neto**  
**Pedro Guilherme Carvalho Silva**

### Instruções Gerais

- A Atividade Complementar deve ser realizada em **dupla**.
- A Atividade Complementar vale até **4,0 pontos**.
- Deverá ser entregue somente **uma única versão por dupla (formato .zip)**.
- A entrega será realizada através de atividade disponibilizada via **Google Classroom**.
- **O arquivo .zip anexado fora do prazo será desconsiderado, de tal modo que será atribuída a nota zero aos alunos.**
- **É de inteira responsabilidade dos alunos a entrega do arquivo .zip para correção.**
- **A solução que não puder ser executada (sem compilar) será desconsiderada durante a correção.**
- **Trabalhos iguais serão automaticamente descartados e os alunos receberão nota zero.**

### Questão única (4,0)

Desenvolver um programa em Java para representar uma biblioteca e seu acervo. Esse programa deve conter as classes Pergamum, AlunoPergamum, LivroPergamum e Main detalhadas abaixo.

- a) **(1,0)** Classe Pergamum responsável por controlar o acervo dos livros dentre outras funcionalidades, para uma determinada biblioteca.

Atributos	Descrição
- codigo int	Código da biblioteca
- nome String	Nome da biblioteca
- listaLivros ArrayList<LivroPergamum>	ArrayList de livros que existem na biblioteca

- listaAlunos ArrayList<AlunoPergamum> ArrayList de alunos que podem pegar livros na biblioteca

Métodos Descrição	
+ Pergamum () Construtor da classe Pergamum vazio.	
+ Pergamum (int código, String nome, int totalLivros, int totalAlunos)	<p>Construtor da classe Pergamum. Após criado o objeto, o código da biblioteca não poderá ter seu código alterado, ou seja, este é o único atributo que não é necessário ter setter. Os parâmetros totalLivros e totalAlunos são os tamanhos dos arraylists listaLivros e listaAlunos, respectivamente.</p> <p>OBS: Todos os setters e getters deverão ser implementados, com exceção do setter de código.</p>
+ boolean cadastrarAluno (AlunoPergamum aluno)	<p>Método responsável por cadastrar um aluno na biblioteca. O método não deverá permitir que um aluno se cadastre mais de uma vez. Caso o aluno informado por parâmetro já exista, deverá ser retornado o valor false.</p>
+ String retirarAluno (int matricula)	<p>Método responsável por retirar um aluno do cadastro da biblioteca. O objeto aluno deverá ser retirado do arrayList. Caso a matrícula informada por parâmetro não exista deverá ser retornado o valor null.</p>
+ boolean cadastrarLivro (LivroPergamum livro)	Método responsável por cadastrar um livro na biblioteca.
+ LivroPergamum retirarLivro (int codigo)	Método responsável por retirar um livro do arrayList de livros.

+ LivroPergamum buscarLivro (int codigo)	Método responsável por buscar um livro no arrayList. Caso o código informado por parâmetro não exista deverá ser retornado o valor null.
------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------

b) **(1,0)** Classe AlunoPergamum responsável por identificar um aluno e realizar operações de empréstimos e devoluções de livros.

Atributos	Descrição
- matricula int	Matrícula do aluno
- nome String	Nome do aluno

- empréstimos ArrayList<LivroPergamum> ArrayList que contém todos os livros que estão emprestados por um aluno

Métodos Descrição	
+ AlunoPergamum () Construtor da classe AlunoPergamum vazio.	
+ Pergamum (int matricula, String nome, int totalLivros)	<p>Construtor da classe AlunoPergamum. O parâmetro totalLivros é a quantidade de livros que o aluno pode pedir emprestado, ou seja, é o tamanho do arrayList empréstimos.</p> <p>OBS: Todos os setters e getters deverão ser implementados.</p>
+ boolean emprestarLivro (LivroPergamum livro)	<p>Método responsável por gerenciar o empréstimo de livros feitos pelo aluno. O livro passado como parâmetro deve ser adicionado ao arrayList empréstimos. O aluno somente pode pedir emprestado 1 (um) exemplar por livro, caso já tenha pedido emprestado um determinado livro e quiser novamente o empréstimo não pode ser realizado. Deve-se verificar se existem livros disponíveis, caso não</p>

	existam, retorna o valor falso.
+ boolean devolverLivro (LivroPergamum livro)	Método responsável por devolver um livro. O objeto livro deverá ser retirado do arrayList empréstimos. Caso o livro informado por parâmetro não exista deverá retornar o valor falso.

c) **(1,0)** Classe LivroPergamum responsável por identificar um livro.

Atributos	Descrição
- codigo int	Código do livro
- titulo String	Título do livro
- autor String	Autor do livro
- ISBN String	ISBN do livro
- totalExemplares int	Número total de exemplares pertencentes a biblioteca deste livro.
- exemplaresEmprestados int	Número de exemplares emprestados do livro na biblioteca.

Métodos	Descrição
+ LivroPergamum ()	Construtor da classe LivroPergamum vazio.
+ LivroPergamum (int codigo, String titulo, String autor, String ISBN, int totalExemplares, int exemplaresEmprestados)	Construtor da classe LivroPergamum. O parâmetro exemplaresEmprestados deverá ser inicializado com 0 (zero).  OBS: Todos os setters e getters deverão ser implementados.
+ int livrosDisponiveis ()	Método responsável por calcular o número de livros que estão disponíveis para novos empréstimos.

+ String imprimirDadosLivro ( )

Método responsável por imprimir as informações do livro como título, autor,

ISBN, número de exemplares existentes e número de exemplares emprestados.

+ boolean isDisponivel ( )

Método responsável por verificar se o empréstimo.

+ void setDisponivel (boolean disponível)  
livro possui exemplares disponíveis para

Método responsável por atualizar o número de exemplares emprestados do livro.

- d) **(1,0)** Classe Main responsável por testar o sistema da biblioteca, isto é, criar objetos para as classes Pergamum (mínimo 1), AlunoPergamum (mínimo 3) e LivroPergamum (mínimo 3). Em seguida, realizar as operações necessárias para o empréstimo e a devolução de livros por parte dos alunos.