

1. Cen rio 1: Uma biblioteca necessita de um software para auxiliar no controle do seu acervo de livros. Para isso:

Crie uma classe **Livro** que contenha os atributos **t tulo**, **g nero**, **autor**, **ISBN**, **ano de publica  o** e possua m todos necess rios para manipular os atributos.

Crie uma classe **Estante** que possua um **c digo** e uma **cole  o de livros**. A classe Estante tamb m deve possuir m todos necess rios para **adicionar** e **excluir** livros em uma estrutura de dados.

Escreva uma classe **Biblioteca** com um m todo **main** que contenha **em torno de 10 livros organizados em estantes**.

-   Simule a remo  o de um dos livros e liste os livros restantes naquela estante.

Sugest o: Utilize a cole  o `java.util.ArrayList` para manipular os dados com mais facilidade.

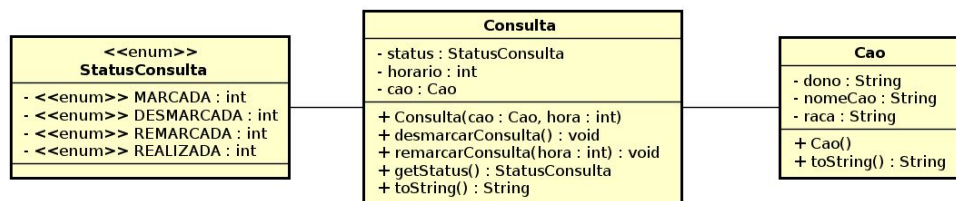
2. Cen rio 2.1: Uma cl nica veterin ria especializada em c es possui uma recepcionista que   encarregada de coletar alguns dados do animal com seu dono, para assim, agendar as consultas. Ela ainda pode:

Remarcar uma consulta existente para outro hor rio;

Desmarcar uma consulta;

Se a consulta ocorreu, ela deve alterar o status da consulta para “realizada”;

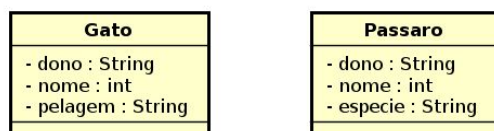
Imagine que voc  foi contratado(a) para informatizar essa cl nica. Codifique um programa em Java com base no diagrama de classes que segue.



powered by Astah

Para a classe **C o**, inclua m todos para alterar e recuperar os valores dos atributos. Para testar o sistema desenvolvido, crie uma classe **C nica** capaz de simular o agendamento de algumas consultas.

3. Cen rio 2.2: Com o passar do tempo, a cl nica veterin ria expandiu-se e passou a atender **gatos** e **p ssaros**, e voc  deve fazer essa atualiza  o no sistema.

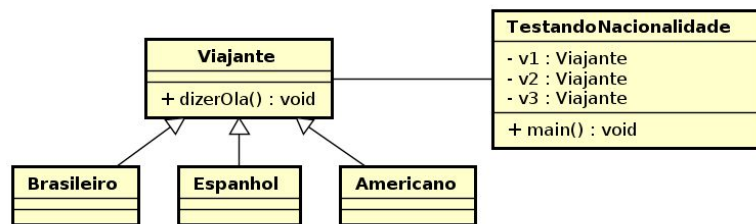


Tarefas:

Adicione as classes Gato e Passaro. Implemente o construtor e os m todos get e set para os atributos;

Aplice o conceito de herança (generalização) para evitar a repetição desnecessária de código;
Cada modelo deve **sobrescrever** o método **toString**;
Altere a classe Consulta fazendo com que a rotina de agendamento (marcação) da consulta seja generalizada, ou seja, sirva para qualquer animal.

4. Considere a modelagem UML:



powered by Astah

Implemente o modelo acima aplicando conceitos de Polimorfismo.

Na classe *TestaNacionalidade*, faça com que cada viajante **diga Olá** em um idioma diferente.

Ex: `v3.dizerOla();`

saída: *Hello*

