

Questão 1 – Classe Mestre

Crie uma classe para representar o **Mestre** do droid de acordo com o diagrama de classe ao lado. Os atributos de um objeto da classe **Mestre** poderão ser setados diretamente (e.g., `mestre1.nome="Poe Dameron"`) ou pelo *método construtor*. Use a técnica de *encadeamento de construtores* para criar os dois construtores da classe, semelhante ao feito em sala e mostrado nos slides. Implemente os métodos:

- `getIdade`: retorna um `int` contendo a idade do mestre em relação ao parâmetro `anoReferencia`. Note que essa idade pode ser negativa (o mestre nasceu **Antes da Batalha de Yavin** -- ABY) ou positiva (nasceu **Depois de Batalha de Yavin** -- DBY).
- `getAnoNascimentoString`: retorna uma `String` contendo o ano de nascimento do mestre seguido da sigla ABY (anoNascimento negativo) ou DBY (anoNascimento positivo).
- `possuiForca`: retorna um `boolean` com o valor `true` caso o atributo `posto` do objeto seja igual a `"Jedi"` ou `"Sith"`.
- `getDescricao`: retorna uma `String` contendo a descrição completa do mestre de acordo com a linha a seguir:

```
Mestre: nome=Luke Skywalker, anoNascimento=19DBY, afiliacao=Alianca  
para Restauracao da Republica, posto=Jedi, possuiForca=true.
```

Para testar a classe, crie uma nova classe chamada **AstromechMain**. Nesta classe, crie o método `main`, que será o ponto de partida do seu programa. No método `main`, crie um ou mais objetos da classe **Mestre** e, em seguida, imprima o resultado da execução do método `getDescricao` dos objetos criados.

Para essa questão, submeta apenas a classe **Mestre**. Não precisa submeter a classe **AstromechMain**.

C	Mestre
a	nome: <i>String</i>
a	anoNascimento: <i>int</i>
a	afiliacao: <i>String</i>
a	posto: <i>String</i>
M	Mestre()
M	Mestre(<i>String</i> nome, <i>int</i> anoNascimento, <i>String</i> afiliacao, <i>String</i> posto)
M	getIdade(<i>int</i> anoReferencia): <i>int</i>
M	possuiForca(): <i>boolean</i>
M	getAnoNascimentoString(): <i>String</i>
M	getDescricao(): <i>String</i>

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Mestre.java"

Solução correta!

Questão 2 – Classe Sensor

2.50 / 2.50

De forma semelhante à questão anterior, implemente uma classe para representar o **Sensor** (câmera) do droid de acordo com o diagrama de classe ao lado. Implemente os métodos:

- `getMPixelsPorSegundo`: retorna um **double** contendo a multiplicação da resolução pela quantidade de frames por segundo.
- `getDescricao`: retorna uma **String** contendo a descrição do sensor de acordo com a linha a seguir:

```
Sensor: cor=Azul, resolucao=512.0Mp, framesPorSegundo=240fps,
mPixelsPorSegundo=122880.0Mpps.
```

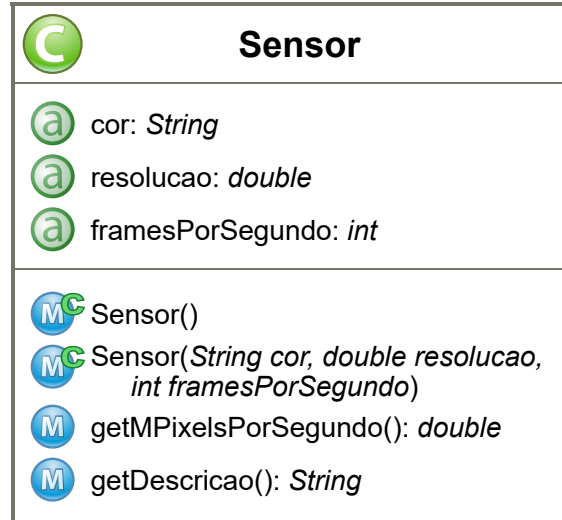
Para testar esta segunda classe, modifique a classe **AstromechMain**, criada na questão anterior para criar um ou mais objetos da classe **Sensor** e, em seguida, imprimir o resultado da execução do método `getDescricao` dos objetos criados.

Para essa questão, submeta apenas a classe **Sensor**. Não precisa submeter a classe **AstromechMain**.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Sensor.java"

Solução correta!

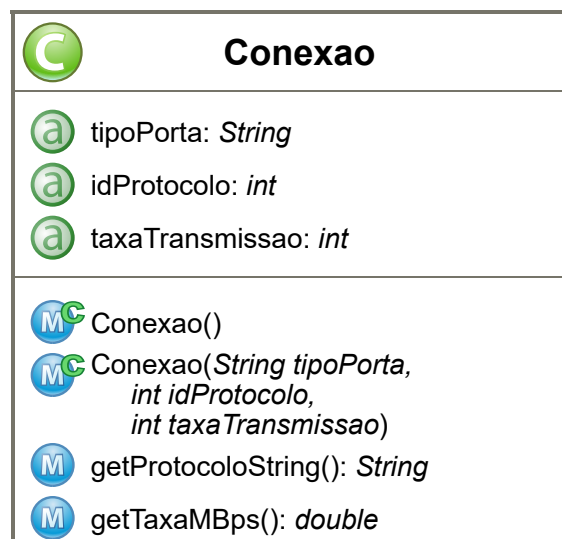


Questão 3 – Classe Conexao

2.00 / 2.00

Como nas questões anteriores, implemente uma classe para representar a **Conexao** do droid de acordo com o diagrama de classe ao lado. Implemente os métodos:

- `getTaxaMBps`: retorna o atributo `taxaTransmissao` (bytes por segundo -- Bps) dividido por 1024 (MBps).



- `getProtocoloString`: retorna o protocolo como `String` de acordo com o atributo `idProtocolo` (1="Rotoscope", 2="Acustico", 3="Radio", qualquer outro id="Outros").
- `getDescricao`: retorna uma `String` contendo a descrição da conexão de acordo com a linha a seguir:

```
Conexao: tipoPorta=SCOMP Link, protocolo=Rotoscope,
taxaTransmissao=4000.0MBps.
```

Para testar esta terceira classe, modifique a classe `AstromechMain`, criada nas questões anteriores para criar um ou mais objetos da classe `Conexao` e, em seguida, imprimir o resultado da execução do método `getDescricao` dos objetos criados.

Para essa questão, submeta apenas a classe `Conexao`. Não precisa submeter a classe `AstromechMain`.

Dica:

- Note que o atributo `taxaTransmissao` é em bytes por segundo (`int`).

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Conexao.java"

Solução correta!

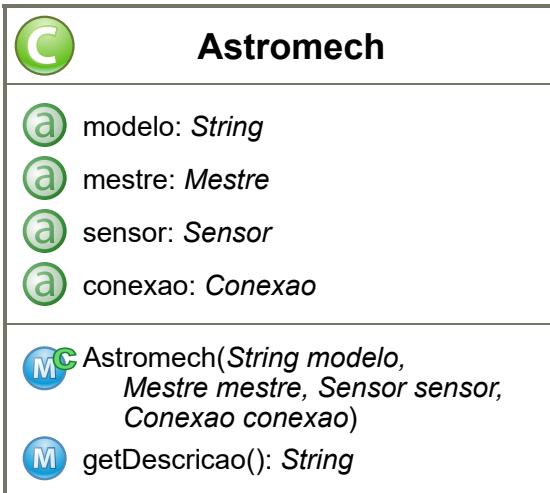
Questão 4 – Classe Astromech

0.00 / 2.50

Implemente uma classe para `Astromech`. Conforme o diagrama de classe ao lado, o droid deverá ter um atributo para seu modelo (`String`), um atributo da classe `Mestre` (criada na Questão 1), um atributo da classe `Sensor` (criada na Questão 2) e um atributo da classe `Conexao` (criada na Questão 3).

Crie o método `getDescricao` que retorne uma `String` contendo todos os dados do droid. Este método deverá se basear na execução do método `getDescricao` dos objetos das outras classes (`Mestre`, `Sensor` e `Conexao`), gerando uma `String` como a seguinte:

```
Astromech modelo R2-D2. Mestre: nome=Luke Skywalker, anoNascimento=19DBY,
afiliacao=Alianca para Restauracao da Republica, posto=Jedi,
possuiForca=true. Sensor: cor=Azul, resolucao=512.0Mp,
framesPorSegundo=240fps, mPixelsPorSegundo=122880.0Mpps. Conexao:
tipoPorta=SCOMP Link, protocolo=Rotoscope, taxaTransmissao=4000.0MBps.
```



Para testar esta classe, modifique a classe **AstromechMain**, criada nas questões anteriores para criar um ou mais objetos da classe **Astromech** e, em seguida, imprimir o resultado da execução do método `getDescricao` dos objetos criados.

Para essa questão, submeta apenas a classe **Astromech**. Não precisa submeter a classe **AstromechMain**.


O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Astromech.java"

Arquivo não compilou. Veja mensagens abaixo.

```
Astromech.java:28: error: no suitable constructor found for
Astromech(String,String,int,String,String,String,double,int,String,int,int)
        Astromech m = new Astromech("R2-D2", "Luke
Skywalker",19 , "Jedi Order", "Jedi","Azul", 512.0, 240,"SCOMP Link", 1,
4096000);
                        ^
        constructor Astromech.Astromech() is not applicable
        (actual and formal argument lists differ in length)
        constructor Astromech.Astromech(String,Mestre,Sensor,Conexao) is not
applicable
        (actual and formal argument lists differ in length)
1 error
```

Questão 5 – Classe AstromechMain

0.00 / 0.50 

Por fim, submeta a classe **AstromechMain**.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AstromechMain.java"