

## **Exercícios propostos**

### **Exercite após o Capítulo 10 – Atividade 1**

#### **Problemas**

Crie os diagramas que representem os itens a seguir.

1. O funcionamento de uma TV simples.
2. Um livro dentro de uma biblioteca.
3. O funcionamento de um carro.
4. A rotina básica de uma pessoa.

# Soluções

Crie os diagramas e as classes em pseudocódigo que representem os itens a seguir.

## 1. TV

Diagrama

TV
MARCA ON_OFF SAP TAMANHO TIPO AV VOLUME CANAL
FABRICADA LIGAR_DESLIGAR MUDAR_AV TECLAR_SAP AUMENTAR_VOLUME DIMINUIR_VOLUME TECLAR_MUDO AUMENTAR_CANAL DIMINUIR_CANAL IR_CANAL

## 2. Livro

Diagrama

LIVRO
TITULO AUTOR EDITORIA ANO PAGINAS ASSUNTO ISBN
EDITAR

## 3. Carro

Diagrama

CARRO
PLACA MARCA FABRICANTE ANO_FABRICA ANO_MODELO CATEGORIA MODELO TAMANHO TIPO TIPO_COMBUSTIVEL CHASSI MARCHA ON_OFF QT_COMBUSTIVEL VELOCIDADE
LIGAR_DESLIGAR ABASTECER AUMENTAR_MARCHA DIMINUIR_MARCHA ACELERAR FREAR PARAR

4. Pessoa

Diagrama

PESSOA
NOME EXISTENCIA {vivo ou morto} ESTADO {acordado ou dormindo} SEXO IDADE ENERGIA FORÇA
NASCER MORRER COMER ANIVERSARIAR DORMIR ACORDAR

## **Exercite após o Capítulo 10 – Atividade 2**

### **Problemas**

Represente os itens a seguir em pseudocódigo das classes propostas na Atividade 1.

1. O funcionamento de uma TV simples.
2. Um livro dentro de uma biblioteca.
3. O funcionamento de um carro.
4. A rotina básica de uma pessoa.

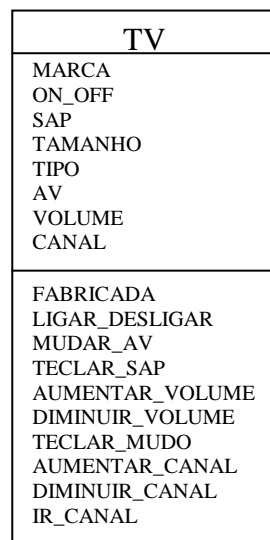
## Soluções

Represente os itens a seguir em pseudocódigo das classes propostas na Atividade 1.

1. O funcionamento de uma TV simples.
2. Um livro dentro de uma biblioteca.
3. O funcionamento de um carro.
4. A rotina básica de uma pessoa.

### 1. TV

#### Diagrama



#### Pseudocódigo

```
classe TV
| MARCA, TIPO: literal
| ON_OFF, AV, SAP: lógico { verdadeiro → ligado, falso → desligado }
| VOLUME, CANAL, TAMANHO: numérico
|
| método FABRICADA (M, T: literal, TAM: numérico)
|   | MARCA ← M
|   | TAMANHO ← T
|   | ON_OFF ← falso
|   | AV ← falso
|   | CANAL ← 0
|   | VOLUME ← 0
|   | SAP ← falso
| fim-método
| método LIGAR_DESLIGAR ( )
|   | se (ON_OFF)
|   |   | então ON_OFF ← falso
|   |   | senão ON_OFF ← verdadeiro
|   fim-se
| fim-método
| método MUDAR_AV ( )
|   | se ( AV )
|   |   | então AV ← falso
|   |   | senão AV ← verdadeiro
|   fim-se
| fim-método
```

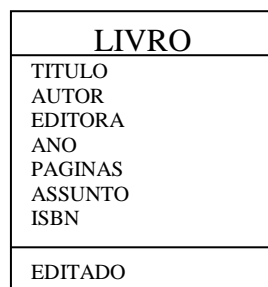
```

| método TECLAR_SAP( )
|   | se ( SAP )
|   |   | então SAP ← falso
|   |   | senão SAP ← verdadeiro
|   | fim-se
| fim-método
| método TECLAR_MUDO( )
|   | VOLUME ← 0
| fim-método
| método numérico AUMENTAR_VOLUME( )
|   | se ( ON_OFF )
|   |   | então VOLUME ← VOLUME + 1
|   |   | fim-se
|   | retorne VOLUME
| fim-método
| método numérico DIMINUIR_VOLUME( )
|   | se ( ON_OFF )
|   |   | então VOLUME ← VOLUME - 1
|   |   | fim-se
|   | retorne VOLUME
| fim-método
| método numérico AUMENTAR_CANAL( )
|   | se ( ON_OFF )
|   |   | então CANAL ← CANAL + 1
|   |   | fim-se
|   | retorne CANAL
| fim-método
| método numérico DIMINUIR_CANAL( )
|   | se ( ON_OFF )
|   |   | então CANAL ← CANAL - 1
|   |   | fim-se
|   | retorne CANAL
| fim-método
| método numérico IR_CANAL(C:numérico)
|   | se ( ON_OFF )
|   |   | então CANAL ← C
|   |   | fim-se
|   | retorne CANAL
| fim-método
fim-classe

```

## 2. Livro

### Diagrama



### Pseudocódigo

```

classe LIVRO
| TITULO,AUTOR,EDITORA,ASSUNTO,ISBN:literal
| ANO,PAGINAS:numérico
|
| método EDITADO (TIT,AUT,ED,ASST,NUMISBN:literal,ANO_ED,NUMPAGS:numérico)

```

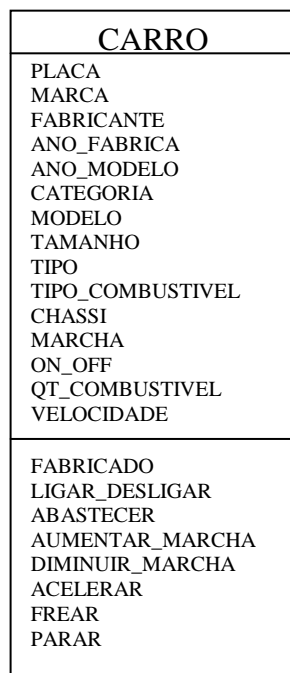
```

| | TITULO ← TIT
| | AUTOR ← AUT
| | EDITORA ← ED
| | ASSUNTO ← ASST
| | ISBN ← NUMISSBN
| | ANO ← ANO_ED
| | PAGINAS ← NUMPAGS
| fim-método
fim-classe

```

### 3. Carro

#### Diagrama



#### Pseudocódigo

classe CARRO

```

| PLACA,MARCA,FABRICANTE,CATEGORIA,MODELO,TIPO,CHASSI:literal
| ON_OFF,FREIO_MAO:lógico { verdadeiro → ligado, falso → desligado }
| ANO_FABRICA,ANO_MODELO,TAMANHO,QT_MARCHA,MARCHA_ATUAL,
| TIPO_COMBUSTIVEL, {0 → flex, 1→gasolina, 2→alcool, 3→ diesel, 4→gas e }
| { 5→ outro}
| QT_COMBUSTIVEL,VELOCIDADE:numérico
|
| método FABRICADO (P,M,F,C,MO,T,CH:literal,ANF,ANM,TAM,QTM,TC:numérico)
|   | PLACA ← P
|   | MARCA ← M
|   | FABRICANTE ← F
|   | CATEGORIA ← C
|   | MODELO ← MO
|   | TIPO ← T
|   | CHASSI ← CH
|   | ON_OFF ← falso
|   | FREIO_MAO ← verdadeiro
|   | ANO_FABRICA ← ANF
|   | ANO_MODELO ← ANM
|   | TAMANHO ← TAM
|   | QT_MARCHA ← QTM
|   | MARCHA_ATUAL ← 0 {ponto morto}

```

```

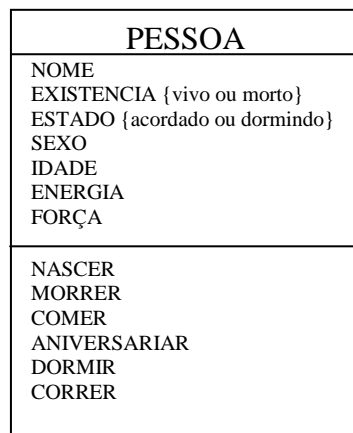
| | TIPO_COMBUSTIVEL ← TC
| | QT_COMBUSTIVEL ← 0
| | VELOCIDADE ← 0
fim-método
método literal LIGAR_DELIGAR( )
| se (ON_OFF)
| | então ON_OFF ← falso
| |     FREIO_MAO ← verdadeiro
| | senão FREIO_MAO ← falso
| |     ON_OFF ← verdadeiro
| fim-se
fim-método
método ABASTECER(QLITROS:numérico)
| se ( ON_OFF )
| | então ON_OFF ← falso
| |     FREIO_MAO ← verdadeiro
| |     MARCHA_ATUAL ← 0
| |     QT_COMBUSTIVEL ← QT_COMBUSTIVEL + QT_LITROS
| fim-se
fim-método
método AUMENTAR_MARCHA( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (MARCHA_ATUAL < QT_MARCHA)
| | | então MARCHA_ATUAL ← MARCHA_ATUAL + 1
| | | senão escreva “Já está na última marcha tolerável para o carro!”
| | fim-se
| | senão escreva “Carro desligado!”
| fim-se
fim-método
método DIMINUIR_MARCHA( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (MARCHA_ATUAL >= 1 )
| | | então MARCHA_ATUAL ← MARCHA_ATUAL - 1
| | | senão escreva “Não é possível diminuir mais a marcha!”
| | fim-se
| | senão escreva “Carro desligado!”
| fim-se
fim-método
método numérico ACELERAR( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (VELOCIDADE < 10)
| | | então VELOCIDADE ← VELOCIDADE + (MARCHA_ATUAL + 1 )
| | | senão escreva “Mude a marcha!”
| | fim-se
| | senão escreva “Carro parado!”
| fim-se
| retorne VELOCIDADE
fim-método
método numérico FREAM( )
| se ( ON_OFF e não FREIO_MAO )
| | então se ( VELOCIDADE > 0 )
| | | então VELOCIDADE ← VELOCIDADE - 5
| | fim-se
| | senão escreva “Carro parado!”
| fim-se
| retorne VELOCIDADE
fim-método
método PARAR( )
| se ( ON_OFF e não FREIO_MAO)
| | então LIGAR_DESLIGAR( )
| |     FREIO_MAO ← verdadeiro
| | fim-se
fim-método
fim-classe

```



## 4. Pessoa

### Diagrama



### Pseudocódigo

classe PESSOA

```
| NOME, SEXO:literal
| EXISTENCIA, { verdadeiro → vivo, falso → morto }
| ESTADO:lógico { verdadeiro → acordado, falso → dormindo }
| IDADE,ENERGIA,FORÇA:numérico

| método NASCER (N,SX:literal)
|   | NOME ← N
|   | SEXO ← SX
|   | EXISTENCIA ← verdadeiro
|   | ESTADO ← verdadeiro
|   | IDADE ← 0 { acabou de nasce }
|   | ENERGIA ← 10 { energia inicial. Vai de 1 a 100. }
|   | FORÇA ← 10 { força. Vai de 1 a 100. }
| fim-método

| método literal MORRER( )
|   | EXISTENCIA ← falso
|   | ESTADO ← falso
|   | ENERGIA ← 0
|   | FORÇA ← 0
|   | retorne "Morto!"
| fim-método

| método COMER( )
|   | se ( EXISTENCIA e FORÇA > 5)
|   |   | então ENERGIA ← ENERGIA + 10
|   |   | FORÇA ← FORÇA + 10
|   |   | senão se (EXISTENCIA)
|   |   |   | então escreva "A pessoa está morta e não pode comer!"
|   |   |   | senão escreva "A pessoa está cansada e precisa dormir um pouco!"
|   |   | fim-se
|   | fim-se
| fim-método

| método ANIVERSARIAR( )
|   | se ( EXISTENCIA)
|   |   | então IDADE ← IDADE+1
|   |   | senão escreva "Pessoa morta não faz aniversário!"
|   | fim-se
| fim-método
```

```

| método DORMIR( )
| | se ( EXISTENCIA )
| | | então ENERGIA ← ENERGIA + 10
| | | FORÇA ← FORÇA + 10
| | | senão escreva "A pessoa está morta!"
| | fim-se
| fim-método
|
| método CORRER( )
| | se ( EXISTENCIA e FORÇA > 5 e ENERGIA > 10 )
| | | então ENERGIA ← ENERGIA - 10
| | | FORÇA ← FORÇA - 5
| | | senão se (EXISTENCIA)
| | | | então escreva "A pessoa está morta e não pode correr!"
| | | | senão se ( ENERGIA < 10 )
| | | | | então escreva "A pessoa está fraca, precisa se alimentar!"
| | | | | senão escreva "A pessoa está exausta, precisa dormir!"
| | | | fim-se
| | | fim-se
| | fim-se
| fim-método
|
fim-classe

```

## **Exercite após o Capítulo 10 – Atividade 3**

### **Problemas**

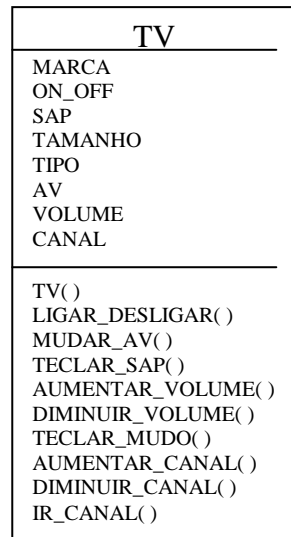
Se necessário, insira os métodos construtores e destruidores e sobrecarregue métodos nos pseudocódigos das classes propostas na Atividade 2.

## Soluções

Se necessário, insira os métodos construtores e destruidores e sobrecarregue métodos nos pseudocódigos das classes propostas na Atividade 2.

### 1. TV

#### Diagrama



#### Pseudocódigo

```
classe TV
| MARCA, TIPO: literal
| ON_OFF, AV, SAP: lógico { verdadeiro → ligado, falso → desligado }
| VOLUME, CANAL, TAMANHO: numérico
|
| construtor TV (M, T: literal, TAM: numérico)
|   | MARCA ← M
|   | TAMANHO ← T
|   | ON_OFF ← falso
|   | AV ← falso
|   | CANAL ← 0
|   | VOLUME ← 0
|   | SAP ← falso
| fim-construtor
| método LIGAR_DESLIGAR ( )
|   | se (ON_OFF)
|   |   | então ON_OFF ← falso
|   |   | senão ON_OFF ← verdadeiro
|   fim-se
| fim-método
| método MUDAR_AV ( )
|   | se ( AV )
|   |   | então AV ← falso
|   |   | senão AV ← verdadeiro
|   fim-se
| fim-método
| método TECLAR_SAP ( )
|   | se ( SAP )
|   |   | então SAP ← falso
|   |   | senão SAP ← verdadeiro
|   fim-se
| fim-método
```

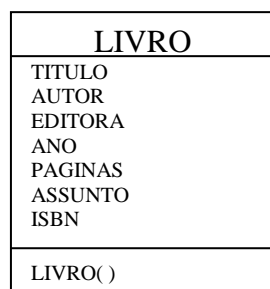
```

| método TECLAR_MUDO( )
|   | VOLUME ← 0
| fim-método
| método numérico AUMENTAR_VOLUME( )
|   | se ( ON_OFF )
|   |   | então VOLUME ← VOLUME + 1
|   | fim-se
|   | retorne VOLUME
| fim-método
| método numérico DIMINUIR_VOLUME( )
|   | se ( ON_OFF )
|   |   | então VOLUME ← VOLUME - 1
|   | fim-se
|   | retorne VOLUME
| fim-método
| método numérico AUMENTAR_CANAL( )
|   | se ( ON_OFF )
|   |   | então CANAL ← CANAL + 1
|   | fim-se
|   | retorne CANAL
| fim-método
| método numérico DIMINUIR_CANAL( )
|   | se ( ON_OFF )
|   |   | então CANAL ← CANAL - 1
|   | fim-se
|   | retorne CANAL
| fim-método
| método numérico IR_CANAL(C:numérico)
|   | se ( ON_OFF )
|   |   | então CANAL ← C
|   | fim-se
|   | retorne CANAL
| fim-método
fim-classe

```

## 2. Livro

### Diagrama



### Pseudocódigo

```

classe LIVRO
| TITULO,AUTOR,EDITOR,A,ASSUNTO,ISBN:literal
| ANO,PAGINAS:numérico
|
| construtor LIVRO (TIT,AUT,ED,ASST,NUMISSBN:literal,ANO_ED,NUMPAGS:numérico)
|   | TITULO ← TIT
|   | AUTOR ← AUT
|   | EDITORA ← ED
|   | ASSUNTO ← ASST
|   | ISBN ← NUMISSBN

```

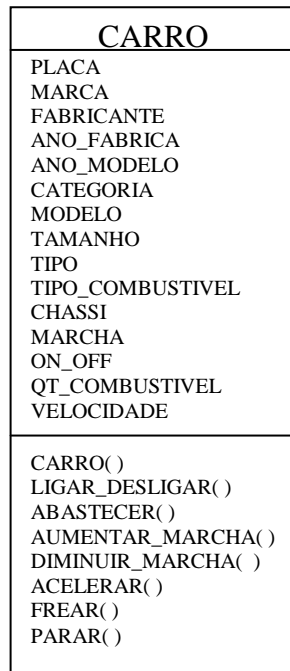
```

| | ANO ← ANO_ED
| | PAGINAS ← Numpags
| fim-construtor
fim-classe

```

### 3. Carro

#### Diagrama



#### Pseudocódigo

classe CARRO

```

| PLACA,MARCA,FABRICANTE,CATEGORIA,MODELO,TIPO,CHASSI:literal
| ON_OFF,FREIO_MAO:lógico { verdadeiro → ligado, falso → desligado }
| ANO_FABRICA,ANO_MODELO,TAMANHO,QT_MARCHA,MARCHA_ATUAL,
| TIPO_COMBUSTIVEL, {0 → flex, 1→gasolina, 2→alcool, 3→ diesel, 4→gas e }
| { 5→ outro}
| QT_COMBUSTIVEL,VELOCIDADE:numérico
|
| construtor CARRO(P,M,F,C,MO,T,CH:literal,ANF,ANM,TAM,QTM,TC:numérico)
|   | PLACA ← P
|   | MARCA ← M
|   | FABRICANTE ← F
|   | CATEGORIA ← C
|   | MODELO ← MO
|   | TIPO ← T
|   | CHASSI ← CH
|   | ON_OFF ← falso
|   | FREIO_MAO ← verdadeiro
|   | ANO_FABRICA ← ANF
|   | ANO_MODELO ← ANM
|   | TAMANHO ← TAM
|   | QT_MARCHA ← QTM
|   | MARCHA_ATUAL ← 0 {ponto morto}
|   | TIPO_COMBUSTIVEL ← TC
|   | QT_COMBUSTIVEL ← 0
|   | VELOCIDADE ← 0
| fim-construtor
| método literal LIGAR_DESLIGAR( )

```

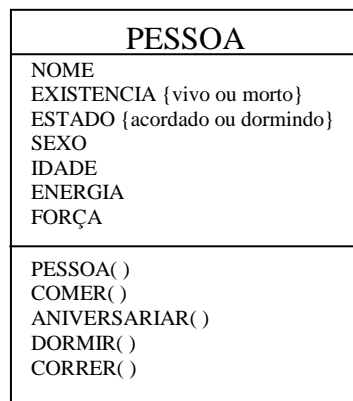
```

| se (ON_OFF)
| | então ON_OFF ← falso
| |     FREIO_MAO ← verdadeiro
| | senão FREIO_MAO ← falso
| |     ON_OFF ← verdadeiro
| fim-se
fim-método
método ABASTECER(QLITROS:numérico)
| se ( ON_OFF )
| | então ON_OFF ← falso
| |     FREIO_MAO ← verdadeiro
| |     MARCHA_ATUAL ← 0
| |     QT_COMBUSTIVEL ← QT_COMBUSTIVEL + QT_LITROS
| fim-se
fim-método
método AUMENTAR_MARCHA( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (MARCHA_ATUAL < QT_MARCHA)
| | | então MARCHA_ATUAL ← MARCHA_ATUAL + 1
| | | senão escreva “Já está na última marcha tolerável para o carro!”
| | fim-se
| | senão escreva “Carro desligado!”
| fim-se
fim-método
método DIMINUIR_MARCHA( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (MARCHA_ATUAL >= 1 )
| | | então MARCHA_ATUAL ← MARCHA_ATUAL - 1
| | | senão escreva “Não é possível diminuir mais a marcha!”
| | fim-se
| | senão escreva “Carro desligado!”
| fim-se
fim-método
método numérico ACELERAR( )
| se ( ON_OFF e não FREIO_MAO )
| | então se (VELOCIDADE < 10)
| | | então VELOCIDADE ← VELOCIDADE + (MARCHA_ATUAL + 1 )
| | | senão escreva “Mude a marcha!”
| | fim-se
| | senão escreva “Carro parado!”
| fim-se
| retorne VELOCIDADE
fim-método
método numérico FREAR( )
| se ( ON_OFF e não FREIO_MAO )
| | então se ( VELOCIDADE > 0 )
| | | então VELOCIDADE ← VELOCIDADE - 5
| | fim-se
| | senão escreva “Carro parado!”
| fim-se
| retorne VELOCIDADE
fim-método
método PARAR( )
| se ( ON_OFF e não FREIO_MAO)
| | então LIGAR_DESLIGAR( )
| |     FREIO_MAO ← verdadeiro
| | fim-se
fim-método
fim-classe

```

## 4. Pessoa

### Diagrama



### Pseudocódigo

classe PESSOA

```
| NOME,SEXO:literal
| EXISTENCIA, { verdadeiro → vivo, falso → morto }
| ESTADO:lógico { verdadeiro → acordado, falso → dormindo }
| IDADE,ENERGIA,FORÇA:numérico

| construtor PESSOA(N,SX:literal)
|   | NOME ← N
|   | SEXO ← SX
|   | EXISTENCIA ← verdadeiro
|   | ESTADO ← verdadeiro
|   | IDADE ← 0 { acabou de nascer }
|   | ENERGIA ← 10 { energia inicial. Vai de 1 a 100. }
|   | FORÇA ← 10 { força. Vai de 1 a 100. }
| fim-construtor

| destruidor PESSOA( )
|   | EXISTENCIA ← falso
|   | ESTADO ← falso
|   | ENERGIA ← 0
|   | FORÇA ← 0
|   | escreva "Morto!"
| fim_destruidor

| método COMER( )
|   | se ( EXISTENCIA e FORÇA > 5)
|   |   | então ENERGIA ← ENERGIA + 10
|   |   | FORÇA ← FORÇA + 10
|   |   | senão se (EXISTENCIA)
|   |   |   | então escreva "A pessoa está morta e não pode comer!"
|   |   |   | senão escreva "A pessoa está cansada e precisa dormir um pouco!"
|   |   | fim-se
|   | fim-se
| fim-método

| método ANIVERSARIAR( )
|   | se ( EXISTENCIA)
|   |   | então IDADE ← IDADE+1
|   |   | senão escreva "Pessoa morta não faz aniversário!"
|   | fim-se
| fim-método
```



```

| método DORMIR( )
| | se ( EXISTENCIA )
| | | então ENERGIA ← ENERGIA + 10
| | | FORÇA ← FORÇA + 10
| | | senão escreva "A pessoa está morta!"
| | fim-se
| fim-método
|
| método CORRER( )
| | se ( EXISTENCIA e FORÇA > 5 e ENERGIA > 10 )
| | | então ENERGIA ← ENERGIA - 10
| | | FORÇA ← FORÇA - 5
| | | senão se (EXISTENCIA)
| | | | então escreva "A pessoa está morta e não pode correr!"
| | | | senão se ( ENERGIA < 10 )
| | | | | então escreva "A pessoa está fraca, precisa se alimentar!"
| | | | | senão escreva "A pessoa está exausta, precisa dormir!"
| | | | fim-se
| | | fim-se
| | fim-se
| fim-método
|
fim-classe

```

## **Exercite após o Capítulo 10 – Atividade 4**

### **Problemas**

Crie uma estrutura de classes utilizando herança que simule:

1. O acervo de uma biblioteca que possui vários itens para empréstimo, como livros, filmes e jogos;
2. A frota de veículos de uma empresa de entrega que possui carros, motos e bicicletas;
3. As pessoas que estão dentro de uma escola, como professores, funcionários, alunos e visitantes.

## Soluções

Crie uma estrutura de classes utilizando herança que simule:

1. O acervo de uma biblioteca que possui vários itens para empréstimo, como livros, filmes e jogos.

Como pode haver objetos que não sejam livros, filmes ou jogos, a classe superior ITEM não pode ser abstrata, pois os objetos que não são criados nas classes inferiores podem ser criados na classe superior.

### Pseudocódigo

classe ITEM

```
| NOME,AUTOR,PRODUTOR,ISBN,TIPO{ ação,comédia,... },DATA_CRIACAO:literal  
| EMPRESTADO, { verdadeiro → sim, falso → não }
```

```
|  
| construtor ITEM(N,A,PR,IS,TP,DT:literal)  
|   | NOME ← N  
|   | AUTOR ← A  
|   | PRODUTOR ← PR  
|   | ISBN ← ISBN  
|   | TIPO ← TP  
|   | DATA_CRIACAO ← DT  
|   | EMPRESTADO ← falso  
| fim-construtor
```

```
| destruidor ITEM()  
|   | escreva "Item descartado do acervo!"  
| fim_destruidor
```

```
| método EMPRESTAR()  
|   | se ( EMPRESTADO )  
|   |   | então escreva "Item já emprestado!"  
|   |   | senão EMPRESTADO ← verdadeiro  
|   | fim-se  
| fim-método
```

```
| método RECEBER()  
|   | se ( EMPRESTADO )  
|   |   | então EMPRESTADO ← falso  
|   |   |   | escreva "Item recebido!"  
|   |   | senão escreva "Item emprestado! Não é este item a ser recebido!"  
|   | fim-se  
| fim-método
```

fim-classe

classe LIVRO herda\_de ITEM

```
| QT_PAGINAS:numérico  
|  
| construtor LIVRO(N,A,PR,IS,DT,TP:literal,QT:numérico)  
|   | super.ITEM(N,A,PR,IS,DT,TP)  
|   | QT_PAGINAS ← QT  
| fim-construtor
```

fim-classe

classe FILME herda\_de ITEM

```
| MIDIA{ VHS,DVD,Blue-ray }:literal  
| DURACAO:numérico  
|  
| construtor FILME(N,A,PR,IS,DT,TP,MD:literal,DU:numérico)  
|   | super.ITEM(N,A,PR,IS,DT,TP)
```

```

|   | MIDIA ← MD
|   | DURACAO ← DU
| fim-construtor
|
fim-classe

classe JOGO herda_de ITEM
| PLATAFORMA { XBOX, Playstation,... }:literal
|
| construtor JOGO(N,A,PR,IS,DT,TP,PL)
|   | super.ITEM(N,A,PR,IS,DT,TP)
|   | PLATAFORMA ← PL
| fim-construtor
|
fim-classe

```

2. A frota de veículos de uma empresa de entrega que possui carros, motos e bicicletas.

Como não pode haver objetos que não sejam carros, motos ou bicicletas, a classe superior VEICULO pode ser abstrata.

### Pseudocódigo

```

classe abstrata VEICULO
| MARCA,FABRICANTE,CATEGORIA,MODELO,COR:literal
| ANO_FABRICA,QT_MARCHA:numérico
|
| método CADASTRAR(M,F,CAT,MO,C:literal,ANF,QTM:numérico)
|   | MARCA ← M
|   | FABRICANTE ← F
|   | CATEGORIA ← CAT
|   | MODELO ← MO
|   | COR ← C
|   | ANO_FABRICA ← ANF
|   | QT_MARCHA ← QTM
| fim-método
|
fim-classe

classe CARRO herda_de VEICULO
| PLACA,TIPO_COMBUSTIVEL, {gasolina,gás, alcool, flex... } CHASSI: literal
| ANO_MODELO:numérico
|
| construtor CARRO(P,M,F,CAT,MO,C,T,CH:literal,ANF,ANM,QTM:numérico)
|   | super.CADASTRAR(M,F,CAT,MO,C,ANF,QTM)
|   | PLACA ← P
|   | TIPO_COMBUSTIVEL ← T
|   | CHASSI ← CH
|   | ANO_MODELO ← ANM
| fim-construtor
|
fim-classe

classe MOTO herda_de CARRO
| QT_CILINDRADA:numérico
| construtor MOTO(P,M,F,CAT,MO,C,T,CH:literal,ANF,ANM,QTM,QTC:numérico)
|   | super.CARRO((P,M,F,CAT,MO,C,T,CH,ANF,ANM,QTM)
|   | QT_CILINDRADA ← QTC
| fim-construtor
|
fim-classe

```

```

classe BICICLETA herda_de VEICULO
| MATERIA_EIXO{ ferro, alumínio... }:literal
| QT_RODAS{ 3-triciclo, 2-normal, 1-monociclo... }:numérico
|
| construtor BICICLETA(M,F,CAT,MO,C,ME:literal,ANF,QTM,QTR:numérico)
|   | super.CADASTRAR(M,F,CAT,MO,C,ANF,QTM)
|   | MATERIA_EIXO ← ME
|   | QT_RODAS ← QTR
|   fim-construtor
fim-classe

```

3. As pessoas que estão dentro de uma escola, como professores, funcionários, alunos e visitantes.

Como os alunos, professores e funcionários são também pessoas, a classe superior pode ser PESSOA, que também servirá para as pessoas visitantes. Portanto, a classe superior PESSOA não pode ser abstrata.

### Pseudocódigo

```

classe PESSOA
| NOME,SEXO,RG:literal
| NUM_CADASTRO:numérico
|
| construtor PESSOA(N,SX,R:literal,NUC:numérico)
|   | NOME ← N
|   | SEXO ← SX
|   | RG ← R
|   | NUM_CADASTRO ← NUC
|   fim-construtor
fim-classe

```

```

classe ALUNO herda_de PESSOA
| TURMA,CURSO,TURNO:literal
| IDADE:numérico
|
| construtor ALUNO(N,SX,R,T,C,TUR:literal,NUC,I:numérico)
|   | super.PESSOA(N,SX,R,NUC)
|   | TURMA ← T
|   | CURSO ← C
|   | TURNO ← TUR
|   | IDADE ← I
|   fim-construtor
fim-classe

```

```

classe FUNCIONARIO herda_de PESSOA
| CPF,CARTEIRA_TRABALHO,FORMACAO,SETOR,CARGO:literal
| IDADE: numérico
|
| construtor FUNCIONARIO(N,SX,R,C,CT,FOR,ST,CA:literal,ID,NUC:numérico)
|   | super.PESSOA(N,SX,R,NUC)
|   | CPF ← C
|   | CARTEIRA_TRABALHO ← CT
|   | FORMACAO ← FOR
|   | SETOR ← ST
|   | CARGO ← CA
|   | IDADE ← ID
|   fim-construtor
fim-classe

```

```
classe PROFESSOR herda_de FUNCIONARIO
| AREA, NIVEL { básico, médio, superior... }: literal
|
| construtor PROFESSOR(N,SX,R,C,CT,FOR,ST,CA,A,NI:literal,ID,NUC:numérico)
|   | super.FUNCIONARIO(N,SX,R,C,CT,FOR,ST,CA,ID,NUC)
|   | AREA ← A
|   | NIVEL ← NI
| fim-construtor
fim-classe
```

## **Exercite após o Capítulo 10 – Atividade 5**

### **Problemas**

Insira os métodos modificadores e implemente o encapsulamento nos códigos feitos na Atividade 4.

## Soluções

1. Insira os métodos modificadores e implemente o encapsulamento nos códigos feitos na Atividade 4.

classe ITEM

```
| privado NOME,AUTOR,PRODUTOR,ISBN,TIPO{ ação,comédia... },DATA_CRIACAO:literal
| privado EMPRESTADO, { verdadeiro → sim, falso → não }

| construtor ITEM(N,A,PR,IS,TP,DT:literal)
|   | NOME ← N
|   | AUTOR ← A
|   | PRODUTOR ← PR
|   | ISBN ← ISBN
|   | TIPO ← TP
|   | DATA_CRIACAO ← DT
|   | EMPRESTADO ← falso
| fim-construtor

| destruidor ITEM( )
|   | escreva "Item descartado do acervo!"
| fim_destruidor

| método público INSERIR_NOME(N:literal)
|   | NOME ← N
| fim-método

| método público INSERIR_AUTOR(A:literal)
|   | AUTOR ← A
| fim-método

| método público INSERIR_PRODUTOR(P:literal)
|   | PRODUTOR ← P
| fim-método

| método público INSERIR_ISBN(IS:literal)
|   | ISBN ← IS
| fim-método

| método público INSERIR_TIPO(T:literal)
|   | TIPO ← T
| fim-método

| método público INSERIR_DATA_CRIACAO(DT:literal)
|   | DATA_CRIACAO ← DT
| fim-método

| método público literal OBTER_NOME( )
|   | retorne NOME
| fim-método

| método público literal OBTER_AUTOR( )
|   | retorne AUTOR
| fim-método

| método público literal OBTER_PRODUTOR( )
|   | retorne PRODUTOR
| fim-método

| método público literal OBTER_ISBN( )
|   | retorne ISBN
| fim-método
```



```

| método público literal OBTER_TIPO( )
|   | retorne TIPO
| fim-método
|
| método público literal OBTER_DATA_CRIACAO( )
|   | retorne DATA_CRIACAO
| fim-método
|
| método público EMPRESTAR( )
|   | se ( EMPRESTADO )
|   |   | então escreva "Item já emprestado!"
|   |   | senão EMPRESTADO ← verdadeiro
|   | fim-se
| fim-método
|
| método público RECEBER( )
|   | se ( EMPRESTADO )
|   |   | então EMPRESTADO ← falso
|   |   |   | escreva "Item recebido!"
|   |   | senão escreva "Item emprestado! Não é este item a ser recebido!"
|   | fim-se
| fim-método
|
fim-classe

```

```

classe LIVRO herda_de ITEM
| privado QT_PAGINAS:numérico
|
| construtor LIVRO(N,A,PR,IS,DT,TP:literal,QT:numérico)
|   | super.ITEM(N,A,PR,IS,DT,TP)
|   | QT_PAGINAS ← QT
| fim-construtor
|
| método público INSERIR_QT_PAGINAS(QT:numérico)
|   | QT_PAGINAS ← QT
| fim-método
|
| método público numérico OBTER_QT_PAGINAS( )
|   | retorne QT_PAGINAS
| fim-método
fim-classe

```

```

classe FILME herda_de ITEM
| privado MIDIA{ VHS,DVD,Blue-ray }:literal
| DURACAO:numérico
|
| construtor FILME(N,A,PR,IS,DT,TP,MD:literal, DU:numérico)
|   | super.ITEM(N,A,PR,IS,DT,TP)
|   | MIDIA ← MD
|   | DURACAO ← DU
| fim-construtor
|
| método público INSERIR_MIDIA(MI:literal)
|   | MIDIA ← MI
| fim-método
|
| método público INSERIR_DURACAO(DU:numérico)
|   | DURACAO ← DU
| fim-método
|
| método público literal OBTER_MIDIA( )
|   | retorne MIDIA
| fim-método
|
| método público numérico OBTER_DURACAO( )
|   | retorne DURACAO

```

```

| fim-método
fim-classe

classe JOGO herda_de ITEM
| PLATAFORMA { XBOX, Playstation... }:literal
|
| construtor JOGO(N,A,PR,IS,DT,TP,PL)
|   | super.ITEM(N,A,PR,IS,DT,TP)
|   | PLATAFORMA ← PL
| fim-construtor
|
| método público INSERIR_PLATAFORMA(PL:literal)
|   | PLATAFORMA ← PL
| fim-método
|
| método público literal OBTER_PLATAFORMA( )
|   | retorne PLATAFORMA
| fim-método
fim-classe

```

## 2. A frota de veículos.

### Pseudocódigo

```

classe abstrata VEICULO
| privado MARCA,FABRICANTE,CATEGORIA,MODELO,COR:literal
| privado ANO_FABRICA,QT_MARCHA:numérico
|
| método CADASTRAR(M,F,CAT,MO,C:literal,ANF,QTM:numérico)
|   | MARCA ← M
|   | FABRICANTE ← F
|   | CATEGORIA ← CAT
|   | MODELO ← MO
|   | COR ← C
|   | ANO_FABRICA ← ANF
|   | QT_MARCHA ← QTM
| fim-método
|
| método público INSERIR_MARCA(M:literal)
|   | MARCA ← M
| fim-método
|
| método público INSERIR_FABRICANTE(F:literal)
|   | FABRICANTE ← F
| fim-método
|
| método público INSERIR_CATEGORIA(C:literal)
|   | CATEGORIA ← C
| fim-método
|
| método público INSERIR_MODELO(M:literal)
|   | MODELO ← M
| fim-método
|
| método público INSERIR_COR(C:literal)
|   | COR ← C
| fim-método
|
| método público INSERIR_ANO_FABRICA(ANF:numérico)
|   | ANO_FABRICA ← ANF
| fim-método
|
| método público INSERIR_QT_MARCHA(QTM:numérico)
|   | QT_MARCHA ← QTM
|

```

```

| fim-método
| método público literal OBTER_MARCA( )
|   | retorne MARCA
| fim-método

| método público literal OBTER_FABRICANTE( )
|   | retorne FABRICANTE
| fim-método

| método público literal OBTER_CATEGORIA( )
|   | retorne CATEGORIA
| fim-método

| método público literal OBTER_MODELO( )
|   | retorne MODELO
| fim-método

| método público literal OBTER_COR( )
|   | retorne COR
| fim-método

| método público numérico OBTER_ANO_FABRICA( )
|   | retorne ANO_FABRICA
| fim-método

| método público numérico OBTER_QT_MARCHA( )
|   | retorne QT_MARCHA
| fim-método
fim-classe

classe CARRO herda_de VEICULO
| PLACA, TIPO_COMBUSTIVEL, { gasolina, gás, alcool, flex... } CHASSI: literal
| ANO_MODELO: numérico

| construtor CARRO(P, M, F, CAT, MO, C, T, CH: literal, ANF, ANM, QTM: numérico)
|   | super.CADASTRAR(M, F, CAT, MO, C, ANF, QTM)
|   | PLACA ← P
|   | TIPO_COMBUSTIVEL ← T
|   | CHASSI ← CH
|   | ANO_MODELO ← ANM
| fim-construtor

| método público INSERIR_PLACA(P: literal)
|   | PLACA ← P
| fim-método

| método público INSERIR_TIPO_COMBUSTIVEL(TC: literal)
|   | TIPO_COMBUSTIVEL ← TC
| fim-método

| método público INSERIR_CHASSI(CH: literal)
|   | CHASSI ← CH
| fim-método

| método público INSERIR_ANO_MODELO(AM: numérico)
|   | ANO_MODELO ← AM
| fim-método

| método público literal OBTER_PLACA( )
|   | retorne PLACA
| fim-método

| método público literal OBTER_TIPO_COMBUSTIVEL( )
|   | retorne TIPO_COMBUSTIVEL
| fim-método

```

```

| método público literal OBTER_CHASSI( )
|   | retorne CHASSI
| fim-método
|
| método público numérico OBTER_ANO_MODELO( )
|   | retorne ANO_MODELO
| fim-método
|
fim-classe

```

```

classe MOTO herda_de CARRO
| QT_CILINDRADA:numérico
| construtor MOTO(P,M,F,CAT,MO,C,T,CH:literal,ANF,ANM,QTM,QTC:numérico)
|   | super.CARRO((P,M,F,CAT,MO,C,T,CH,ANF,ANM,QTM)
|   | QT_CILINDRADA ← QTC
| fim-construtor
|
| método público INSERIR_QT_CILINDRADA(QT:numérico)
|   | QT_CILINDRADA ← QT
| fim-método
|
| método público numérico OBTER_QT_CILINDRADA( )
|   | retorne QT_CILINDRADA
| fim-método
|
fim-classe

```

```

classe BICICLETA herda_de VEICULO
| MATERIA_EIXO{ ferro, alumínio... }:literal
| QT_RODAS{ 3-triciclo, 2-normal, 1-monociclo... }:numérico
|
| construtor BICICLETA(M,F,CAT,MO,C,ME:literal,ANF,QTM,QTR:numérico)
|   | super.CADASTRAR(M,F,CAT,MO,C,ANF,QTM)
|   | MATERIA_EIXO ← ME
|   | QT_RODAS ← QTR
| fim-construtor
|
| método público INSERIR_MATERIA_EIXO(ME:literal)
|   | MATERIA_EIXO ← ME
| fim-método
|
| método público INSERIR_QT_RODAS(QTR:numérico)
|   | QT_RODAS ← QTR
| fim-método
|
| método público literal OBTER_MATERIA_EIXO( )
|   | retorne MATERIA_EIXO
| fim-método
|
| método público numérico OBTER_QT_RODAS( )
|   | retorne QT_RODAS
| fim-método
|
fim-classe

```

3. As pessoas que estão dentro de uma escola, como professores, funcionários, alunos e visitantes.

### Pseudocódigo

```
classe PESSOA
| NOME,SEXO,RG:literal
| NUM_CADASTRO:numérico
|
| construtor PESSOA(N,SX,R:literal,NUC:numérico)
|   | NOME ← N
|   | SEXO ← SX
|   | RG ← R
|   | NUM_CADASTRO ← NUC
| fim-construtor
|
| método público INSERIR_NOME(N:literal)
|   | NOME ← N
| fim-método
|
| método público INSERIR_SEXO(SX:literal)
|   | SEXO ← SX
| fim-método
|
| método público INSERIR_RG(R:literal)
|   | RG ← R
| fim-método
|
| método público INSERIR_NUM_CADASTRO(NUC:numérico)
|   | NUM_CADASTRO ← NUC
| fim-método
|
| método público literal OBTER_NOME( )
|   | retorne NOME
| fim-método
|
| método público literal OBTER_SEXO( )
|   | retorne SEXO
| fim-método
|
| método público literal OBTER_RG( )
|   | retorne RG
| fim-método
|
| método público numérico OBTER_NUM_CADASTRO( )
|   | retorne NUM_CADASTRO
| fim-método
|
fim-classe
```

```
classe ALUNO herda_de PESSOA
| TURMA,CURSO,TURNO:literal
| IDADE:numérico
|
| construtor ALUNO(N,SX,R,T,C,TUR:literal,NUC,I:numérico)
|   | super.PESSOA(N,SX,R,NUC)
|   | TURMA ← T
|   | CURSO ← C
|   | TURNO ← TUR
|   | IDADE ← I
| fim-construtor
|
| método público INSERIR_TURMA(T:literal)
|   | TURMA ← T
```

```

| fim-método

| método público INSERIR_CURSO(C:literal)
|   | CURSO ← C
| fim-método

| método público INSERIR_TURNO(T:literal)
|   | TURNO ← T
| fim-método

| método público INSERIR_IDADE(ID:numérico)
|   | IDADE ← ID
| fim-método

| método público literal OBTER_TURMA( )
|   | retorne TURMA
| fim-método

| método público literal OBTER_CURSO( )
|   | retorne CURSO
| fim-método

| método público literal OBTER_TURNO( )
|   | retorne TURNO
| fim-método

| método público literal OBTER_IDADE( )
|   | retorne IDADE
| fim-método
fim-classe

classe FUNCIONARIO herda_de PESSOA
| CPF,CARTEIRA_TRABALHO,FORMACAO,SETOR,CARGO:literal
| IDADE: numérico

| construtor FUNCIONARIO(N,SX,R,C,CT,FOR,ST,CA:literal,ID,NUC:numérico)
|   | super.PESSOA(N,SX,R,NUC)
|   | CPF ← C
|   | CARTEIRA_TRABALHO ← CT
|   | FORMACAO ← FOR
|   | SETOR ← ST
|   | CARGO ← CA
|   | IDADE ← ID
| fim-construtor

| método público INSERIR_CPF(C:literal)
|   | CPF ← C
| fim-método

| método público INSERIR_CARTEIRA_TRABALHO(CT:literal)
|   | CARTEIRA_TRABALHO ← CT
| fim-método

| método público INSERIR_FORMACAO(F:literal)
|   | FORMACAO ← F
| fim-método

| método público INSERIR_SETOR(ST:literal)
|   | SETOR ← ST
| fim-método

| método público INSERIR_CARGO(CA:literal)
|   | CARGO ← CA
| fim-método

| método público INSERIR_IDADE(ID:numérico)

```

```

    | IDADE ← ID
fim-método

método público literal OBTER_CPF( )
    | retorne CPF
fim-método

método público literal OBTER_CARTEIRA_TRABALHO( )
    | retorne CARTEIRA_TRABALHO
fim-método

método público literal OBTER_FORMACAO( )
    | retorne FORMACAO
fim-método

método público literal OBTER_SETOR( )
    | retorne SETOR
fim-método

método público literal OBTER_CARGO( )
    | retorne CARGO
fim-método

método público numérico OBTER_IDADE( )
    | retorne IDADE
fim-método
fim-classe

classe PROFESSOR herda_de FUNCIONARIO
    AREA, NIVEL { básico, médio, superior... }:literal

    construtor PROFESSOR(N,SX,R,C,CT,FOR,ST,CA,A,NI:literal,ID,NUC:numérico)
        | super.FUNCIONARIO(N,SX,R,C,CT,FOR,ST,CA,ID,NUC)
        | AREA ← A
        | NIVEL ← NI
    fim-construtor

    método público INSERIR_AREA(A:literal)
        | AREA ← A
    fim-método

    método público INSERIR_NIVEL(N:literal)
        | NIVEL ← N
    fim-método

    método público literal OBTER_AREA( )
        | retorne AREA
    fim-método

    método público literal OBTER_NIVEL( )
        | retorne NIVEL
    fim-método
fim-classe

```

## Exercite após o Capítulo 10 – Atividade 6

### Problemas

Seja uma biblioteca que tem um acervo de 10 mil itens divididos em livros, revistas, audiovisuais, jogos e outros itens diversos. Cada tipo possui os dados a seguir.

- Item diverso:
  - informações a serem armazenadas:
    - nome, autor, produtor, área de conhecimento, data de criação e ISBN.
- Livro
  - informações a serem armazenadas:
    - nome, autor, editora, área de conhecimento, data de criação, ISBN, número de páginas e número de edição.
- Revista
  - informações a serem armazenadas:
    - nome, editor, editora, área de conhecimento, data de criação, ISSN, número de páginas, ano de edição e número do volume.
- Audiovisual
  - informações a serem armazenadas:
    - nome, diretor, produtor, área de conhecimento, data de criação, ISBN, mídia, duração, ator principal e sinopse.
- Jogo
  - informações a serem armazenadas:
    - nome, autor, produtor, área de conhecimento, data de criação, ISBN, mídia e plataforma.

Você foi contratado para elaborar um sistema que faça o cadastro e a consulta desse acervo. Como você acabou de estudar Orientação a Objetos, implemente uma solução simples que use os conceitos de OO.



## Soluções

Seja uma biblioteca que tem um acervo de 10 mil itens divididos em livros, revistas, audiovisuais, jogos e outros itens diversos. Cada tipo possui os dados a seguir.

- Item diverso:
  - informações a serem armazenadas:
    - nome, autor, produtor, área de conhecimento, data de criação e ISBN.
- Livro
  - informações a serem armazenadas:
    - nome, autor, editora, área de conhecimento, data de criação, ISBN, número de páginas e número de edição.
- Revista
  - informações a serem armazenadas:
    - nome, editor, editora, área de conhecimento, data de criação, ISSN, número de páginas, ano de edição e número do volume.
- Audiovisual
  - informações a serem armazenadas:
    - nome, diretor, produtor, área de conhecimento, data de criação, ISBN, mídia, duração, ator principal e sinopse.
- Jogo
  - informações a serem armazenadas:
    - nome, autor, produtor, área de conhecimento, data de criação, ISBN, mídia e plataforma.

Você foi contratado para elaborar um sistema que faça o cadastro e consulta desse acervo. Como você acabou de estudar Orientação a Objetos, implemente uma solução simples que use os conceitos de OO.

### Explicação inicial

Como resposta a esta atividade, utilizaremos um vetor de 10 mil posições. A utilização de arquivos em orientação a objetos necessitaria de mais um capítulo para explicar suas particularidades. Infelizmente não temos como fazer isso, pois sairíamos de um capítulo introdutório a OO.

Outro detalhe importante, é que o vetor será de objetos ITEM, pois, caso necessitemos inserir outro tipo, não haverá problema porque todos são descendentes de ITEM e podem ser inseridos no vetor.

classe ITEM

```
| privado NOME,AUTOR,PRODUTOR,AREA,ISBN,DATA_CRIACAO:literal
|
| construtor ITEM(N,A,PR,AR,IS,DT:literal)
|   | NOME ← N
|   | AUTOR ← A
|   | PRODUTOR ← PR
|   | ISBN ← IS
|   | AREA ← AR
|   | DATA_CRIACAO ← DT
| fim-construtor
|
| método público INSERIR_NOME(N:literal)
|   | NOME ← N
| fim-método
|
| método público INSERIR_AUTOR(A:literal)
|   | AUTOR ← A
| fim-método
|
| método público INSERIR_PRODUTOR(P:literal)
|   | PRODUTOR ← P
| fim-método
|
```

```

| método público INSERIR_ISBN(IS:literal)
|   | ISBN ← IS
| fim-método
|
| método público INSERIR_AREA(A:literal)
|   | AREA ← A
| fim-método
|
| método público INSERIR_DATA_CRIACAO(DT:literal)
|   | DATA_CRIACAO ← DT
| fim-método
|
| método público literal OBTER_NOME( )
|   | retorne NOME
| fim-método
|
| método público literal OBTER_AUTOR( )
|   | retorne AUTOR
| fim-método
|
| método público literal OBTER_PRODUTOR( )
|   | retorne PRODUTOR
| fim-método
|
| método público literal OBTER_ISBN( )
|   | retorne ISBN
| fim-método
|
| método público literal OBTER_AREA( )
|   | retorne AREA
| fim-método
|
| método público literal OBTER_DATA_CRIACAO( )
|   | retorne DATA_CRIACAO
| fim-método
|
| método público literal MOSTRA_DADOS( )
|   | retorne "Item: ",NOME," Autor: ",AUTOR," Produtor: ",PRODUTOR," ISBN: ",ISBN,
|   | "Área: ",AREA," Data de criação:",DATA_CRIACAO
| fim-método
|
fim-classe

```

classe LIVRO herda\_de ITEM

```

| privado QT_PAGINAS,NUM_EDICAO:numérico
|
| construtor LIVRO(N,A,PR,AR,IS,DT:literal,QT,NE:numérico)
|   | super.ITEM(N,A,PR,AR,IS,DT)
|   | QT_PAGINAS ← QT
|   | NUM_EDICAO ← NE
| fim-construtor
|
| método público INSERIR_QT_PAGINAS(QT:numérico)
|   | QT_PAGINAS ← QT
| fim-método
|
| método público INSERIR_NUM_EDICAO(NE:numérico)
|   | NUM_EDICAO ← NE
| fim-método
|
| método público numérico OBTER_QT_PAGINAS( )
|   | retorne QT_PAGINAS
| fim-método
|
| método público numérico OBTER_NUM_EDICAO( )
|   | retorne NUM_EDICAO

```

```

| fim-método
|
| método público literal MOSTRA_DADOS( )
|   | retorne "Livro: ",NOME," Autor: ",AUTOR," Editora: ",PRODUTOR," ISBN: ",ISBN,
|   |   "Área: ",AREA," Data de criação:",DATA_CRIACAO," Quantidade de páginas: ",QT,
|   |   " Número da edição:",NUM_EDICAO
| fim-método
|
fim-classe

```

```

classe REVISTA herda_de LIVRO
| privado NUM_VOLUME:numérico
|
| construtor REVISTA(N,A,PR,AR,IS,DT:literal,QT,NE,NV:numérico)
|   | super.LIVRO(N,A,PR,AR,IS,DT,QT,NE)
|   | NUM_VOLUME ← NV
| fim-construtor
|
| método público INSERIR_NUM_VOLUME(NV:numérico)
|   | NUM_VOLUME ← NV
| fim-método
|
| método público numérico OBTER_NUM_VOLUME( )
|   | retorne NUM_VOLUME
| fim-método
|
| método público literal MOSTRA_DADOS( )
|   | retorne "Revista: ",NOME," Editor: ",AUTOR," Editora: ",PRODUTOR," ISSN: ",ISSN,
|   |   "Área: ",AREA," Data de criação:",DATA_CRIACAO," Quantidade de páginas: ",QT,
|   |   " Ano de edição:",NUM_EDICAO,"Número do volume: ",NUM_VOLUME
| fim-método
|
fim-classe

```

```

classe AUDIOVISUAL herda_de ITEM
| privado MIDIA{ VHS,DVD,Blue-ray },ATOR,SINOPSE:literal
| privado DURACAO:numérico
|
| construtor AUDIOVISUAL(N,A,PR,AR,IS,DT,MD,AT,SIN:literal,DU:numérico)
|   | super.ITEM(N,A,PR,AR,IS,DT)
|   | MIDIA ← MD
|   | ATOR ← AT
|   | SINOPSE ← SIN
|   | DURACAO ← DU
| fim-construtor
|
| método público INSERIR_MIDIA(MI:literal)
|   | MIDIA ← MI
| fim-método
|
| método público INSERIR_ATOR(AT:literal)
|   | ATOR ← AT
| fim-método
|
| método público INSERIR_SINOPSE(SIN:literal)
|   | SINOPSE ← SIN
| fim-método
|
| método público INSERIR_DURACAO(DU:numérico)
|   | DURACAO ← DU
| fim-método
|
| método público literal OBTER_MIDIA( )
|   | retorne MIDIA
| fim-método
|

```

```

| método público literal OBTER_ATOR( )
|   | retorne ATOR
| fim-método
|
| método público literal OBTER_SINOPSE( )
|   | retorne SINOPSE
| fim-método
|
| método público numérico OBTER_DURACAO( )
|   | retorne DURACAO
| fim-método
|
| método público literal MOSTRA_DADOS( )
|   | retorne "Audiovisual: ",NOME," Diretor: ",AUTOR," Produtora: ",PRODUTOR," ISBN: ",
|   |         ISBN," Área: ",AREA," Data de criação:",DATA_CRIACAO," Mídia: ",MIDIA,
|   |         " Ator principal:",ATOR," Sinopse: ",SINOPSE," Duração:",DURACAO
| fim-método
|
fim-classe

```

classe JOGO herda\_de ITEM

```

| privado MIDIA,PLATAFORMA { XBOX,Playstation,... }:literal
|
| construtor JOGO(N,A,PR,AR,IS,DT,MI,PL)
|   | super.ITEM(N,A,PR,AR,IS,DT)
|   | MIDIA ← MI
|   | PLATAFORMA ← PL
| fim-construtor
|
| método público INSERIR_PLATAFORMA(PL:literal)
|   | PLATAFORMA ← PL
| fim-método
|
| método público INSERIR_MIDIA(MI:literal)
|   | MIDIA ← MI
| fim-método
|
| método público literal OBTER_PLATAFORMA( )
|   | retorne PLATAFORMA
| fim-método
|
| método público literal OBTER_MIDIA( )
|   | retorne MIDIA
| fim-método
|
| método público literal MOSTRA_DADOS( )
|   | retorne "Jogo: ",NOME," Autor: ",AUTOR," Produtora: ",PRODUTOR," ISBN: ",
|   |         ISBN," Área: ",AREA," Data de criação:",DATA_CRIACAO," Mídia: ",MIDIA,
|   |         " Plataforma:",PLATAFORMA
| fim-método
|
fim-classe

```

classe BIBLIOTECA

```

| público ACERVO: conjunto [1..10000] ITEM
| privado ULTIMAPOSICAO:numérico
|
| construtor BIBLIOTECA( )
|   | ULTIMAPOSICAO ← 1
| fim-construtor
|
| método público INSERIR_ULTPOS(UP:numérico)
|   | ULTIMAPOSICAO ← UP
| fim-método
|
| método público numérico OBTER_ULTPOS( )

```

```

| | retorne ULTIMAPOSICAO
fim-método

método público CADASTRAR( )
| NM,ATR,PRDTR,AR,ISS,DTC,MD,AT,SIN,PL:literal
| NP,NE,NV,DU:numérico
| OPMENU:numérico
| se (OBTER_ULTPOS( ) > 10000)
| | então escreva "Acervo lotado! Não é possível inserir mais item!"
| | senão repita
| | | escreva "Escolha a opção de item para cadastro"
| | | escreva " 1 – Livro"
| | | escreva " 2 – Revista"
| | | escreva " 3 – Audiovisual (DVD ou VHS)"
| | | escreva " 4 – Jogo"
| | | escreva " 5 – Outro item"
| | | escreva " 0 – Sair"
| | | repita
| | | | leia OPMENU
| | até que ((OPMENU=1) ou (OPMENU=2) ou (OPMENU=3) ou
| | | (OPMENU=4) ou (OPMENU = 5) ou (OPMENU = 0))
| | caso OPMENU
| | | seja 1 faça leia "Digite o nome do livro:", NM
| | | | leia "Digite o nome do autor do livro:",ATR
| | | | leia "Digite o nome do editor do livro:",PRDTR
| | | | leia "Digite a área de conhecimento do livro:",AR
| | | | leia "Digite o ISBN do livro:",ISS
| | | | leia "Digite a data de criação do livro (DD/MM/AA):",DTC
| | | | leia "Digite o número de páginas do livro:",NP
| | | | leia "Digite o número da edição do livro:",NE
| | | | ACERVO[OBTER_ULTPOS( )] ←
| | | | | novo LIVRO(NM,ATR,PRDTR,AR,ISS,DTC,NP,NE)
| | | | INSERIR_ULTPOS(OBTER_ULTPOS( )+1)
| | | | seja 2 faça leia "Digite o nome da revista:", NM
| | | | | leia "Digite o nome do editor-chefe da revista:",ATR
| | | | | leia "Digite o nome da editora da revista:",PRDTR
| | | | | leia "Digite a área de conhecimento da revista:",AR
| | | | | leia "Digite o ISSN da revista:",ISS
| | | | | leia "Digite a data de criação da revista (DD/MM/AA):",DTC
| | | | | leia "Digite o número de páginas da revista:",NP
| | | | | leia "Digite o ano de edição da revista:",NE
| | | | | leia "Digite o número do volume da revista:",NV
| | | | | ACERVO[OBTER_ULTPOS( )] ←
| | | | | novo REVISTA(NM,ATR,PRDTR,AR,ISS,DTC,NP,NE,NV)
| | | | | INSERIR_ULTPOS(OBTER_ULTPOS( )+1)
| | | | seja 3 faça leia "Digite o nome do audiovisual:", NM
| | | | | leia "Digite o nome do diretor do audiovisual:",ATR
| | | | | leia "Digite o nome da produtora do audiovisual:",PRDTR
| | | | | leia "Digite a área de conhecimento do audiovisual:",AR
| | | | | leia "Digite o ISBN do audiovisual:",ISS
| | | | | leia "Digite a data de criação (DD/MM/AA):",DTC
| | | | | leia "Digite a duração em minutos:",DU
| | | | | leia "Digite o tipo de mídia (DVD,VHS...):",MD
| | | | | leia "Digite o nome do(a) ator/atriz principal:",AT
| | | | | leia "Digite a sinopse do audiovisual:",SIN
| | | | | ACERVO[OBTER_ULTPOS( )] ←
| | | | | novo AUDIOVISUAL(NM,ATR,PRDTR,AR,ISS,DTC,MD,
| | | | | | AT,SIN,DU)
| | | | | INSERIR_ULTPOS(OBTER_ULTPOS( )+1)
| | | | seja 4 faça leia "Digite o nome do jogo:", NM
| | | | | leia "Digite o nome do autor do jogo:",ATR
| | | | | leia "Digite o nome da produtora do jogo:",PRDTR
| | | | | leia "Digite a área de conhecimento do jogo:",AR
| | | | | leia "Digite o ISSBN do jogo:",ISS
| | | | | leia "Digite a data de criação (DD/MM/AA):",DTC
| | | | | leia "Digite o tipo de mídia (DVD,CD...):",MD

```

```

leia "Digite o nome da plataforma do jogo: ", PL
ACERVO[OBTER_ULTPOS( )] ←
novo JOGO(NM,ATR,PRDTR,AR,ISS,DTC,MD,PL)
INSERIR_ULTPOS(OBTER_ULTPOS( )+1)
seja 5 faça leia "Digite o nome do item: ", NM
leia "Digite o nome do autor do item: ", ATR
leia "Digite o nome da produtora do item: ", PRDTR
leia "Digite a área de conhecimento do item: ", AR
leia "Digite o ISBN do item: ", ISS
leia "Digite a data de criação (DD/MM/AA): ", DTC
ACERVO[OBTER_ULTPOS( )] ←
novo ITEM(NM,ATR,PRDTR,AR,ISS,DTC)
INSERIR_ULTPOS(OBTER_ULTPOS( )+1)
fim-caso
até que (OPMENU = 0)
fim-se
fim-método

método público EXCLUIR(NOME:literal)
| ACHOU:lógico
| POSPESQ, POSMOV:numérico
| POSPESQ ← 1
| ACHOU ← falso
| enquanto (( POSPESQ <= OBTER_ULTPOS( ) e (não ACHOU))
| | se ACERVO[POSPESQ].OBTER_NOME( ) = NOME
| | | então para POSMOV de POSPESQ + 1 até OBTER_ULTPOS( ) faça
| | | | ACERVO[POSPESQ] ← ACERVO[POSMOV]
| | | | POSPESQ ← POSPESQ + 1
| | | fim-para
| | | ACHOU ← verdadeiro
| | | escreva "Item excluído!"
| | | senão POSPESQ ← POSPESQ + 1
| | fim-se
| fim-enquanto
| se não ACHOU
| | então escreva "Este item não consta no acervo!"
| fim-se
fim-método

método público literal CONSULTAR(NOME:literal)
| ACHOU:lógico
| POSPESQ:numérico
| POSPESQ ← 1
| ACHOU ← falso
| enquanto (( POSPESQ <= OBTER_ULTPOS( ) e (não ACHOU))
| | se ACERVO[POSPESQ].OBTER_NOME( ) = NOME
| | | então ACHOU ← verdadeiro
| | | senão POSPESQ ← POSPESQ + 1
| | fim-se
| se não ACHOU
| | então retorne "Este item não consta no acervo!"
| | senão retorne ACERVO[POSPESQ].MOSTRA_DADOS( )
| fim-se
fim-método

início
| BIBLIOTECA BIBLIOTECASENAC:novo BIBLITECA( )
| OPCA:numérico
| repita
| | escreva "Digite o número da opção desejada:"
| | escreva " 1 – Cadastro"
| | escreva " 2 – Exclusão"
| | escreva " 3 – Consulta"
| | escreva " 0 – Sair"
| | repita
| | | leia OPCA

```

```
| | até que ((OPCAO = 1) ou (OPCAO = 2) ou (OPCAO = 3) ou (OPCAO = 0))
| | caso OPCA0
| |   seja 1 faça BIBLIOTECASENAC.CADASTRAR( )
| |   seja 2 faça leia "Digite o nome do item que deseja excluir:",NOMEITEM
| |               BIBLIOTECASENAC.EXCLUIR(NOMEITEM)
| |   seja 3 faça leia "Digite o nome do item que deseja consultar:",NOMEITEM
| |               escreva BIBLIOTECASENAC.CONSULTAR(NOMEITEM)
| |   seja 4 faça escreva "Obrigado por nos visitar!"
| | fim-caso
| até que (OPCAO = 4)
| fim
fim-classe
```