

Exercício de revisão - BIBLIOTECA



A folha de exercícios foi desenvolvida para ser resolvida por módulos. Os módulos foram pensados para serem resolvidos de maneira incremental e precisam ser resolvidos em ordem.

!

Módulo 01: POO + Swift

Parte 1:

1. Crie um diagrama de classes que modele um objeto Livro. Lembre-se de que ele NÃO representa uma cópia específica de um livro, mas da obra de forma abstrata. Como regra geral, um livro deve ter nome (**String**), um código ISBN (**Int**) e um autor (**String**).
2. Implemente a classe criando os atributos e construtores que forem necessários.
3. No Playground, crie três objetos Livro.

Parte 2:

1. Crie um diagrama de classe que represente um modelo de sócio de uma biblioteca. Como regra geral, um sócio deve ter um nome (**String**), um sobrenome (**String**) e um número de identificação (**Int**).
2. Implemente a classe criando os atributos e construtores que forem necessários.
3. No Playground, crie três objetos Sócio.

Parte 3:

Queremos adicionar ao modelo anterior uma nova categoria de sócios: os sócios VIP. Além de um nome, um sobrenome e um número de identificação, estes sócios têm um valor de mensalidade (**Int**).

1. Como você modificaria o diagrama de sócio criado anteriormente?
2. Modifique a implementação considerando os novos requisitos. Crie as classes que forem necessárias.
3. Crie os atributos necessários.
4. No Playground, crie dois objetos SócioVIP.

**BACK
FRONT**

Módulo 01: POO + Swift

Parte 4:

BACK FRONT

1. Crie um diagrama de classes com um modelo de exemplar da biblioteca. Lembre-se de que um exemplar representa uma cópia e uma edição particulares de um livro específico.

Como regra geral, um exemplar deve ter um Livro (**Livro**), um número de edição (**Int**), uma localização (**String**) e um número de identificação (**Int**).

2. Implemente a classe criando os atributos necessários.

**BACK
FRONT**

3. No Playground, crie exemplares diferentes para cada Livro criado.



!

Módulo 01: POO + Swift

Parte 5:

Além de ter nome, código ISBN e autor, o livro de uma biblioteca possui uma lista de exemplares (**[Exemplar]**) disponíveis para empréstimo.

BACK FRONT

1. Como você modificaria o diagrama de livro criado anteriormente?
2. Modifique a implementação considerando os novos requisitos.



Módulo 01: POO + Swift

Parte 6:

Além de ter nome, sobrenome e número de identificação, um sócio de uma biblioteca possui uma lista de exemplares retirados (**[Exemplar]**) e uma quantidade máxima de livros que podem ser retirados. (**Int**). Um sócio comum pode retirar até 3 livros. Por outro lado, um sócio VIP pode retirar até 15 livros.

1. Como você modificaria os diagramas de sócio e sócio VIP criados anteriormente?



2. Modifique a implementação considerando os novos requisitos.

Parte 7:

1. Na classe Livro, crie um método que permita adicionar um novo exemplar de livro à lista de exemplares disponíveis.
 - **adicionar(umExemplar : Exemplar) -> Void**
2. Na classe Livro, crie um método que permita consultar se há exemplares disponíveis de um livro. Se houver exemplares disponíveis na lista, este método deverá retornar true. Caso contrário, deverá retornar false.
 - **temExemplaresDisponíveis() -> Bool**
3. Na classe Livro, crie um método que permita emprestar um exemplar de um livro. Este método deve remover o primeiro exemplar da lista de exemplares disponíveis (assumindo que a lista NÃO está vazia) e retorná-lo.
 - **removerExemplar() -> Exemplar**
4. Na classe Livro, crie um método que permita registrar o recebimento de um exemplar que foi emprestado a um sócio. Este método deve adicionar o primeiro exemplar recebido como parâmetro à lista de exemplares disponíveis.

Módulo 01: POO + Swift

- **receber(umExemplar : Exemplar) -> Void**

5. No Playground, adicione os Exemplares criados aos seus Livros correspondentes.

Módulo 01: POO + Swift

Parte 8:

1. Na classe Sócio, crie um método que permita consultar se um sócio tem capacidade disponível para retirar um livro. Este método retorna true se tiver capacidade (ou seja, menos exemplares retirados do que os permitidos) e false se não tiver capacidade.
Atenção: Lembre-se de que um sócio comum pode retirar apenas três exemplares, enquanto um sócio VIP pode retirar até 15 livros.
 - **temCapacidadeDisponível() -> Bool**
2. Na classe Sócio, crie um método que permita que o sócio pegue emprestado um exemplar. Isso significa que o método deverá adicionar o exemplar enviado por parâmetro à lista de exemplares retirados do sócio.
 - **pegarEmprestado(umExemplar : Exemplar) -> Void**

Módulo 01: POO + Swift

3. Na classe Sócio, crie um método que permita devolver um exemplar. Isso significa que o método deverá eliminar o exemplar recebido por parâmetro da lista de exemplares retirados, já que ele foi devolvido pelo sócio.
 - **devolver(umExemplar : Exemplar) -> Void**

Parte 9:

Agregue ao modelo anterior uma representação do objeto Empréstimo, que representa o empréstimo de um exemplar a um sócio. Como regra geral, o objeto deve ter um exemplar (**Exemplar**), um sócio (**Sócio**) e uma data (**Data**). Não é necessário registrar a data de término do empréstimo.

1. Modifique o diagrama de classes e modele a classe Empréstimo.
2. Implemente a classe criando os atributos necessários.
3. Crie um construtor que tome o sócio e o exemplar como parâmetro e construa um empréstimo com a data do dia.

*A classe Date permite utilizar datas em Swift. Para criar a data do dia basta instanciar um novo objeto usando **Date()**.*

Parte 10:

Por último, queremos adicionar uma representação da Biblioteca ao modelo anterior. Uma biblioteca tem uma lista de livros (**[Livro]**), uma lista de sócios (**[Sócio]**) e uma lista de empréstimos efetuados (**[Empréstimo]**).

1. Modifique o diagrama de classes para que modele o objeto Biblioteca.
2. Implemente a classe criando os atributos necessários.

Módulo 01: POO + Swift

Parte 11:

1. Crie um método na classe Biblioteca que permita registrar um sócio.
 - **registrar(sócio: Sócio) -> Void**
2. Crie um método na classe Biblioteca que permita registrar um livro.
 - **registrar(livro:Livro) -> Void**
3. Na classe Biblioteca, crie um método que permita emprestar um exemplar do livro solicitado pelo sócio. O resultado da operação deve ser impresso na tela. ◦ **emprestar(isbnLivro: Int, idSócio: Int) -> Void**

Em primeiro lugar, o método deve:

- Procurar o livro na lista de livros da biblioteca (usando o ISBN enviado por parâmetro). Armazená-lo em uma variável.
- Procurar o sócio na lista de sócios (usando o número de identificação enviado por parâmetro). Armazená-lo em uma variável.

Em seguida, ele deverá conferir:

- Se o livro tem exemplares disponíveis.
- Se o sócio tem capacidade disponível.

Em caso afirmativo, ou seja, se as condições anteriores forem verdadeiras, deve-se:

- Procurar um exemplar do livro a emprestar. *Usar os métodos já definidos em Livro.*
- Registrar que o sócio retirou esse exemplar. *Usar os métodos já definidos em Sócio.*
- Criar um objeto Empréstimo, carregar nele as informações necessárias e adicioná-lo ao registro de empréstimos da Biblioteca.

4. Crie um método na classe Biblioteca que permita registrar a devolução de um exemplar. O resultado da operação deve ser impresso na tela.
 - **retornar (umExemplar: Exemplar, idSócio: Int) -> Void**

O método deve:

Módulo 01: POO + Swift

- Percorrer a lista de sócios, comparando se o número de identificação passado por parâmetro corresponde ao número de identificação de algum sócio na lista. Assim que o sócio for encontrado, registrar que devolveu o exemplar.
- Percorrer a lista de sócios, comparando se o livro do exemplar passado por parâmetro corresponde a algum da lista. Assim que o livro correspondente ao exemplar for encontrado, registrar o recebimento do exemplar.

5. No Playground, crie um objeto Biblioteca e registrar os Sócios e os Livros criados.

Parte 12:

1. Na classe Biblioteca, crie um método que permita emprestar um exemplar de cada livro da lista de ISBN solicitada pelo sócio.

- **emprestar(listaDoSBNs: [Int], idSócio: Int) -> Void**

Para cada ISBN da lista, o método deve:

- Procurar o livro na lista de livros da biblioteca usando o ISBN enviado por parâmetro. Armazená-lo em uma variável.
- Procurar o sócio na lista de sócios usando o número de identificação enviado por parâmetro. Armazená-lo em uma variável.

Em seguida, ele deverá conferir:

- Se o livro tem exemplares disponíveis.
- Se o sócio tem capacidade disponível.

Em caso afirmativo, ou seja, se as condições anteriores forem verdadeiras, deve-se:

- Procurar um exemplar do livro a emprestar.
Usar os métodos já definidos em Livro.
- Registrar que o sócio retirou esse exemplar.
Usar os métodos já definidos em Sócio.
- Criar um objeto Empréstimo, carregar nele as informações necessárias e adicioná-lo ao registro de empréstimos da Biblioteca.

Imprima na tela o resultado do empréstimo.

Módulo 01: POO + Swift

ESCLARECIMENTO: Lembre-se dos métodos realizados anteriormente. O sócio levará os exemplares dos livros que estiverem disponíveis. Caso haja um exemplar de um livro que não possa ser retirado, essa informação será exibida na tela.

2. Crie um método que permita registrar a devolução de uma lista de exemplares. Esse método não retorna nada.

- **retornar(listaDeExemplares: [Exemplar], idSócio: Int) -> Void**

Para cada exemplar, o método deve:

- Registrar que o sócio devolveu o exemplar.
- Registrar que o exemplar foi recebido na biblioteca.

Imprima na tela o resultado do empréstimo.

3. No Playground, simule empréstimos e devoluções.

Exercício adicional

Parte 13

A Biblioteca está incorporando uma classificação de seus livros por categoria, para mostrar ao usuário a lista de livros disponíveis em cada categoria.

1. Crie um diagrama de classes com o modelo do objeto Categoria. Como regra geral, uma categoria deve ter um nome (**String**), um código (**Int**) e uma descrição (**String**).
2. Implemente a classe Categoria definindo os atributos necessários.
3. Modifique a classe Biblioteca da seguinte forma:
 - adicionando uma lista de categorias, onde todas são registradas.
 - adicionando um dicionário com chaves, que serão os códigos das categorias, e valores, que serão uma lista de livros associados a essa categoria.
4. Implemente os métodos a seguir:

Módulo 01: POO + Swift

- **registrar(categoria: Categoria) -> Void** que deve adicionar a categoria recebida por parâmetro à lista de categorias.
- **adicionar(umLivro: Livro, a umaCategoria: Categoria) -> Void** que deve receber como parâmetros uma categoria e um livro, e adicionar o livro ao dicionário com a categoria correspondente. Se a categoria não existir, crie a categoria e a lista ao adicionar o livro.
- **listarLivros(de umaCategoria: Categoria) -> [Livro]** que deve receber como parâmetro uma categoria e retornar uma lista com todos os livros que pertencem a essa categoria.
- **informarCategoria(de umLivro: Livro) -> Categoria?** que deve receber um livro como parâmetro e retornar a categoria a qual ele pertence.

Parte 14

A biblioteca define que, além de Livros, também pretende emprestar exemplares de **Revistas** e de **Artigos**. Queremos adicionar ao modelo anterior dois tipos de publicações que serão emprestadas junto com os livros.

1. Modifique o diagrama de classes para que modele os objetos **Artigo** e **Revista**.
 - Um artigo tem um ISBN, uma lista de autores, uma data de publicação, um resumo e uma lista de exemplares que podem ser emprestados.
 - Uma revista tem um ISBN, um autor, uma data de publicação e uma lista de exemplares a serem emprestados.
2. Implemente as duas classes criando os atributos necessários.
3. Implemente a herança utilizando como classe pai "**Publicação**".
 - a. Modifique a classe Exemplar para que sua referência seja uma Publicação.
 - b. Modifique a Biblioteca para que permita realizar empréstimos de Publicações em geral. Como você poderá ver, é necessário modificar vários métodos que estavam programados. No entanto, o que antes chamávamos de Livro agora

Módulo 01: POO + Swift

é uma Publicação, portanto, a coisa mais simples seria renomear a classe Livro como Publicação e, em seguida, criar uma nova classe Livro.

Módulo 01: POO + Swift

Parte 15

A biblioteca define que quer imprimir seus próprios exemplares de livros e artigos, mas não de revistas. A biblioteca quer receber uma lista de livros e artigos para poder imprimir um exemplar de cada um deles.

1. Modifique o diagrama de classe para que represente o modelo do protocolo Imprimível, que contém o seguinte método:
 - **imprimir() -> Void**

Este método é responsável por criar um exemplar do livro ou do artigo com os seguintes dados:

- número de edição 2017 (ou seja, ano atual).
- localização "A DEFINIR".
- id -1.

Este método deverá adicionar o exemplar à lista de exemplares do livro ou artigo.

2. Modifique as classes Livro e Artigo para que implementem a interface.
3. Adicione à classe Biblioteca o método a seguir:
 - **imprimirNovosExemplares(imprimíveis: [Imprimível])** que deve receber uma lista de imprimíveis como parâmetro e imprimir um exemplar de cada.