Objetos em Swift

**Antes de implementar qualquer exercício, é necessário desenvolver o diagrama de classes correspondente.**

**Exercício 1:**

1. Definir uma nova classe**Cliente** contendo os seguintes atributos:

* Nome
* Sobrenome

1. Definir uma nova classe **Conta** contendo os seguintes atributos:

* Número da conta
* Saldo
* Titular (o titular é um objeto do tipo Cliente).

1. Criar o construtor da classe **Cliente**, usado para configurar o nome o sobrenome. Criar também o construtor da classe **Conta** que configure o número de conta e o titular (o saldo sempre deve começar zerado).
2. Criar um método chamado **depositar** que tenha como parâmetro de entrada **quantia em dinheiro**. Como estamos fazendo um depósito, a quantia de dinheiro é somada ao saldo. O método deve imprimir na tela o tipo de transação realizada e o novo saldo.
3. Criar um método chamado **extrair** que tenha como parâmetro de entrada **quantia em dinheiro**. Como estamos fazendo um saque, a quantia é subtraída do saldo. Caso o valor do saque a realizar seja maior que o saldo disponível, imprimir na tela: “Saldo insuficiente”. Caso contrário, o método deve imprimir na tela o tipo de transação realizada e o novo saldo.
4. Escrever um programa que crie duas pessoas, configurando os atributos necessários, e criar uma conta para cada uma, também com os atributos necessários. Em seguida, fazer um depósito e um saque em cada conta.

**Exercício 2:**

1. Definir uma nova classe **JogadorDeFutebol** contendo os seguintes atributos, com os valores por defeito indicados:
   * Nome : “Jogador”
   * Energia : 100
   * Alegria : 50
   * Gols : 0
   * Experiência : 0
2. Definir uma nova classe **SessaoDeTreinamento** contendo os seguintes atributos:
   * Experiência ganha quando um treinamento é realizado
3. Definir o construtor de **SessaoDeTreinamento** que receba como parâmetro a quantidade de pontos de experiência que serão designados em cada treinamento.
4. Criar um método chamado **fazerGol** na classe **JogadorDeFutebol**, que tire 5 de energia e dê 10 de alegria ao Jogador. O método deve imprimir na tela “GOOOOL!” e acrescentar 1 aos gols do jogador.
5. Criar um método chamado **correr** na classe **JogadorDeFutebol**, que tire 10 energia do Jogador. O método deve imprimir na tela: “Cansei”.
6. Criar um método chamado **treinarA** na classe **SessaoDeTreinamento** que tenha como parâmetro de entrada **um jogador de futebol**. Ao ser treinado, o jogador precisa:
   * Correr
   * Fazer um Gol
   * Correr

Depois que o jogador fizer essas três coisas, a experiência dele deve aumentar, somando os pontos de experiência do treinamento.  
O método deve imprimir a experiência inicial e final do jogador.

1. Escrever um programa que crie dois jogadores de futebol, configurando os atributos necessários, e que crie uma sessão de treinamento, também com os atributos necessários. Depois, cada jogador deve realizar um treinamento.
2. Modificar o sistema acrescentando um construtor à classe **JogadorDeFutebol**.

**Exercício 3:**

1. Definir uma nova classe**Atleta** contendo os seguintes atributos:

* Nome
* Nível
* Energia

1. Definir uma nova classe **Prova** contendo os seguintes atributos:

* Dificuldade Necessária
* Energia Necessária

1. Criar, na classe **Prova,** o método **podeRealizar,** que tenha como parâmetro de entrada **um objeto do tipo Atleta.** O método deve retornar **True** caso o atleta possa realizar a prova e, **caso contrário,** False.

Um atleta pode realizar a prova se tiver um nível maior ou igual à dificuldade da prova, além de energia suficiente.

1. Escrever um programa que crie dois atletas, configurando os atributos necessários, e criar três provas para cada um, também com os atributos necessários. Depois, verificar se os atletas podem realizar essas provas.

**Exercício 4:**

O **tripé** é um aparelho de três pés, com a parte superior circular ou triangular, que permite estabilizar uma câmera e evitar os movimentos dela. Modelar o comportamento desse objeto.

1. Definir uma classe **Tripé** contendo os seguintes atributos:
   * Dobrado: indica o estado atual do tripé (dobrado ou não)
   * Altura mínima
   * Altura máxima
   * Altura atual
2. Definir o método **definirAltura(novaAltura: Int)**, que receba uma altura e modifique o valor da altura atual.
3. Definir o método **dobrar()**, que permita dobrar o tripé.
4. Definir o método **desdobrar()**, que permita desdobrar o tripé.
5. Definir o método **guardar()**, que permita deixar o tripé pronto para guardar. Ou seja, ele deve estar dobrado e a altura atual deve ser a menor possível.
6. Definir o método **prontoParaGuardar?()**, que permita verificar se o tripé está pronto para ser guardado. Ou seja, ele deve estar dobrado e com a mínima altura atual.
7. Definir o método **usar()**, que permita usar o tripé. Ou seja, ele deve estar desdobrado e a altura atual deve ser a maior possível.
8. Definir o método **prontoParaUsar?()**, que permita verificar se o tripé está pronto para ser usado. Ou seja, ele deve estar desdobrado e a altura atual deve ser a maior possível.
9. Escrever um programa que crie um tripé e faça testes com os métodos definidos anteriormente.

**Exercício 5:**

Desenvolver um sistema informático para uma concessionária de veículos usados. Essa concessionária decidiu que venderá apenas carros. Todos os carros têm marca, modelo, ano de fabricação, cor e quilometragem.

Como estratégia de marketing, a concessionária quer ter um registro dos clientes que compraram os veículos. O departamento de marketing da concessionária precisa saber o nome, o sobrenome e dados de contato dos clientes para poder oferecer outros carros no futuro.

Por outro lado, o departamento de contabilidade da concessionária quer ter um registro das vendas realizadas. Cada registro deve ter as seguintes informações:

* Valor da venda
* Veículo vendido
* Cliente

1. Fazer um diagrama de classe representando o modelo que queremos implementar.
2. Implementar cada classe e definir os atributos. As classes que devem ser implementadas são: Concessionária, Cliente, Venda e Veículo.
3. Criar os initializers para as classes Cliente, Veículo e Venda.
   1. Cliente terá como parâmetros: nome, sobrenome e contato.
   2. Veículo terá como parâmetros: marca, modelo, ano de fabricação, cor e quilometragem.
   3. Venda terá como parâmetros: cliente, veículo e valor da venda.
4. Definir, na classe Concessionária, o método **registrarVenda(veículo: Veículo, cliente: Cliente, valor: Double)**, que acrescenta ao registro de vendas da concessionária uma nova venda de um veículo, cliente por um valor determinado.

Em um Playground, executar:

criar um carro, criar um cliente e registrar a venda desse veículo a esse cliente pela Concessionária.

## 

## 

## ADICIONAL

**Exercício 6 (Atletas 2.0 reloaded):**

Estender o exercício dos atletas para que:

* Um atleta possa competir em um dos estádios do Rio. Cada estádio tem um conjunto de provas.
* Dado um atleta, queremos saber que provas ele pode realizar em um estádio.
  + O método recebe um atleta e retorna uma lista com as provas que podem ser feitas no estádio.
* Dados dois atletas e uma prova, queremos saber quem é o melhor. Ou seja, o atleta que concluir a prova e que tiver o maior nível atlético.
* Dados dois atletas e um estádio, definir qual atleta ganhará a medalha de ouro. Ou seja, aquele que conseguir realizar melhor o maior número de provas.