



## Revisão - exercícios

**Obs: Use os conceitos de classes e construtores aprendidos em sala de aula para a resolução dos exercícios.**

### Exercício 1

A Digital House decidiu desenvolver um sistema de cadastro de alunos com seus respectivos cursos, professores responsáveis e turmas, para esse sistema foram levantados os seguintes requisitos:

- A matéria dada em cada aula possui um nome;
- A aula possui uma matéria, um horário de início e término;
- O aluno possui um RA (Registro Acadêmico), nome, sobrenome.
- O aluno pode assistir as aulas e fazer as lições de casa;
- O curso possui um nome, uma lista de aulas, uma lista de alunos e um professor responsável;
- O professor possui um nome e um RD (Registro Docente);
- O professor por ser um Docente tem a obrigatoriedade de dar aulas e fazer a chamada dos alunos;
- A turma deverá conter um nome e um curso associado.



## Revisão - exercícios

### Exercício 2

Crie uma classe chamada Fatura que possa ser utilizado por uma loja de suprimentos de informática para representar uma fatura de itens vendidos na loja. Uma fatura deve incluir as seguintes informações como atributos:

- Uma lista de itens cada Item possui:

- o número do item faturado;
- a descrição do item;
- a quantidade comprada do item
- o preço unitário do item.

Sua classe Item deve conter um construtor que inicialize os quatro atributos. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0.

Além disso, na classe Fatura uma função chamada totalDaFatura que calcula o valor da fatura (isso é, multiplicar a quantidade pelo preço de cada item) e depois retorna o valor como um Double.

Escreva um arquivo com uma função Main de teste que demonstra os comportamentos da classe Fatura.



## Revisão - exercícios

### Exercício 3

Escreva uma classe Estoque. Ela deverá possuir:

- os atributos nome (String), qtdAtual (Int) e qtdMinima (Int);
- as funções com as seguintes assinaturas:

```
fun mudarNome(String nome): Unit
```

```
fun mudarQtdMinima(Int qtdMinima): Unit
```

```
fun repor(Int qtd): Unit
```

```
fun darBaixa(Int qtd): Unit
```

```
fun mostra(): Unit
```

```
fun precisaRepor(): Boolean
```

Os atributos qtdAtual e qtdMinima jamais poderão ser negativos.

- A função repor aumenta qtdAtual de acordo com o parâmetro qtd;
- A função darBaixa diminui qtdAtual de acordo com o parâmetro qtd;
- A função mostra() retorna uma String contendo o nome do produto, sua quantidade mínima, sua quantidade atual;
- A função precisaRepor retorna true caso a quantidade atual esteja menor ou igual à quantidade mínima e false, caso contrário.