



Revisão - exercícios

Obs: Use os conceitos de construtores classe abstrata, interfaces, herança e entrada de dados dinâmicas aprendidas 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.

Escreva a função main de teste que demonstra as capacidades do sistema da DigitalHouse.





Revisão - exercícios



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 propriedades:

- Uma lista de itens cada `Item` possui:
 - 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 as quatro propriedades. 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 chamado `getTotalFatura` 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 a função `main` de teste que demonstra as capacidades da classe `Fatura`.



Revisão - exercícios

Exercício 3

Escreva uma classe Estoque. Ela deverá possuir:

- as propriedades: nome, qtdAtual e qtdMinima;
- as funções com as seguintes assinaturas:

mudarNome(nome: String)

mudarQtdMinima(qtdMinima: Int)

repor(qtd: Int)

darBaixa (qtd: Int)

mostra(): String

precisaRepor(): boolean

As propriedades 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.

Escreva a função main de teste que demonstra as capacidades da classe Estoque.



Revisão - exercícios