

## PRÁTICA LABORATORIAL 13

### Objetivos:

- Interfaces

## EXERCÍCIOS

### Parte 1

1. Crie uma classe abstrata chamada "Animal" com o método abstrato:
  - i. void emitirSom();
  - b. Crie duas interfaces: "Aquatico" e "Terrestre" com os métodos:
    - i. void nadar() (para "Aquatico")
    - ii. void andar() (para "Terrestre")
  - c. Crie quatro classes que estendem a classe abstrata "Animal" e implementam as interfaces "Aquatico" e "Terrestre": "Peixe", "Baleia", "Cao" e "Gato".
  - d. Na classe "Peixe", implemente o método "emitirSom()" para exibir "Glub glub!" e o método "nadar()" para exibir "A nadar...".
  - e. Na classe "Cao", implemente o método "emitirSom()" para exibir "Au au!" e o método "andar()" para exibir "A andar...".
  - f. Na classe "Baleia", implemente o método "emitirSom()" para exibir "Eeeeeeeooooo!" e o método "nadar()" para exibir "A nadar...".
  - g. Na classe "Gato", implemente o método "emitirSom()" para exibir "Miau!" e o método "andar()" para exibir "A andar...".
2. Crie uma classe abstrata chamada "Veiculo" com os atributos "marca" e "ano" e os métodos abstratos:
  - i. void acelerar();
  - ii. void frear();
  - b. Crie duas classes que estendem a classe abstrata "Veiculo": "Carro" e "Mota".
  - c. Na classe "Carro", implemente os métodos para exibir "Carro a acelerar..." e "Carro a travar...".
  - d. Na classe "Mota", implemente os métodos para exibir "Mota a acelerar..." e "Mota a travar...".
  - e. Crie uma interface chamada "Transporte" com o método:
    - i. void transportar();
  - f. Implemente a interface "Transporte" nas classes "Carro" e "Moto".
  - g. Na classe "Carro", implemente o método para exibir "A transportar passageiros de carro...".
  - h. Na classe "Mota", implemente o método para exibir "A transportar passageiro de mota...".

- 
3. Crie uma classe abstrata chamada "ContaBancaria" com os atributos "titular" e "saldo" e os métodos abstratos:
    - i. `void depositar(double valor);`
    - ii. `void levantar(double valor);`
    - iii. `void mostrarSaldo();`
  - b. Crie duas classes que estendem a classe abstrata "ContaBancaria": "ContaCorrente" e "ContaPoupanca".
  - c. Na classe "ContaCorrente", implemente os métodos para atualizar o saldo após um depósito ou levantamento e imprima o saldo atualizado.
  - d. Na classe "ContaPoupanca", implemente os métodos para atualizar o saldo após um depósito ou levantamento e exiba o saldo atualizado.
  - e. Crie uma interface chamada "Transferivel" com o método:
    - i. `void transferir(double valor, ContaBancaria contaDestino);`
  - f. Implemente a interface "Transferivel" na classe "ContaCorrente".
  - g. No método "transferir", verifique se o saldo é suficiente para realizar a transferência e atualize os saldos das contas de origem e destino.

4. Crie uma classe abstrata chamada "Pessoa" com os atributos "nome" e "idade" e os métodos abstratos:
  - i. `abstract void exibirDados();`
- b. Crie uma classe abstrata chamada "Paciente" que estende a classe abstrata "Pessoa" e tem um atributo adicional "numeroPaciente".
- c. Implemente o método "exibirDados()" para exibir o nome, idade e número do paciente.
- d. Crie uma interface chamada "Interna" com os métodos:
  - i. `void internar();`
  - ii. `void realizarExames();`
- e. Crie uma classe chamada "Medico" que estende a classe abstrata "Pessoa" e implementa a interface "Interna".
- f. Implemente os métodos "exibirDados()", "internar()" e "realizarExames()".
- g. No método "exibirDados()", exiba o nome e idade do médico.
- h. No método "internar()" imprima na consola "A internar o paciente..."
- i. No método "realizarExames()" recebe um ArrayList de Strings com os exames a serem realizados e imprime na consola "A realizar [Exame 1]..." "A realizar [Exame 2]...", por exemplo, "A realizar RX..." "A realizar TAC..."
- j. Crie uma classe chamada "Hospital" que possui um método estático (de classe) chamado "atenderPaciente".
- k. O método "atenderPaciente" recebe um objeto do tipo "Paciente" como parâmetro e chama o método "exibirDados()" do paciente. O método "atenderPaciente" também recebe um objeto do tipo "Medico" como parâmetro e chama os métodos "exibirDados()", "internar()" e "realizarExames()" do médico. Assim como outros argumentos que considere relevante.
- l. No main, instancie dois pacientes e um médico. De seguida, invoque o método "atenderPaciente" como por exemplo `Hospital.atenderPaciente(paciente1, medico1);`
- m. Crie uma interface chamada "Cirurgiao" com o método:
  - i. `void realizarCirurgia();`
- n. No método "realizarCirurgia()", recebe como parâmetro uma String com a parte do corpo e imprime na consola "A realizar uma cirurgia: [parte do corpo]..."
- o. Crie uma classe `MedicoCirurgiao` que estende a classe `Medico` e tenha como atributo adicional a especialidade. Esta classe deve implementar a interface `Cirurgiao`.
- p. Na classe `hospital` implemente o método "operarPaciente", por sua vez recebe como parâmetro o paciente e o `medicoCirurgiao`, assim como outros argumentos que considere relevante. Este método deve invocar o método de `atenderPaciente` e ainda o método de `realizarCirurgia`.
- q. No main, acrescente a instancia deste cirurgião, e instancie um novo paciente a ser operado, invoque o método `operarPaciente`.

**Bom trabalho! 😊**