



## Lista 15 - Classes - Introdução à programação orientada a objetos.

### 1. (Exercício de aquecimento)

Crie uma classe em C++ para representar uma pessoa. A classe deve conter os seguintes atributos privados: nome (string), idade (int) e altura (float). Implemente os seguintes métodos públicos:

- Construtor que receba nome, idade e altura;
- Métodos acessores (get) e modificadores (set) para cada atributo — também chamados de métodos de acesso;
- Um método para imprimir todos os dados da pessoa.

No programa principal, faça o seguinte:

- Crie um objeto `pessoa1` utilizando o construtor, passando os dados diretamente;
- Crie um objeto `pessoa2`, leia os dados do teclado (via `cin`) e atribua-os usando os métodos `set`;
- Imprima os dados de ambas as pessoas utilizando o método de impressão da classe.

*Após ler o código apresentado como exemplo, identifique onde cada item solicitado no enunciado está sendo implementado na solução (atributos, construtor, métodos `get/set`, método de impressão e criação dos objetos).*

### Solução em C++:

```
#include <iostream>
#include <string>
using namespace std;

class Pessoa {
private:
    string nome;
    int idade;
    float altura;

public:
    // Construtor
    Pessoa(string nome, int idade, float altura) {
        this->nome = nome;
        this->idade = idade;
        this->altura = altura;
    }
}
```

```

// Construtor padrao (opcional)
Pessoa() {
    nome = "";
    idade = 0;
    altura = 0.0;
}

// Metodos set
void setNome(string nome) {
    this->nome = nome;
}

void setIdade(int idade) {
    this->idade = idade;
}

void setAltura(float altura) {
    this->altura = altura;
}

// Metodos get
string getNome() {
    return nome;
}

int getIdade() {
    return idade;
}

float getAltura() {
    return altura;
}

// Metodo para imprimir os dados
void imprimirDados() {
    cout << "Nome: " << nome << endl;
    cout << "Idade: " << idade << " anos" << endl;
    cout << "Altura: " << altura << " metros" << endl;
}
};

int main() {
    // Criando pessoa1 com construtor
    Pessoa pessoa1("Joao", 25, 1.75);

    // Criando pessoa2 e preenchendo com set
    Pessoa pessoa2;
    string nome;
    int idade;
    float altura;

    cout << "Digite o nome da pessoa2: ";

```

```

getline(cin, nome);
pessoa2.setNome(nome);

cout << "Digite a idade da pessoa2: ";
cin >> idade;
pessoa2.setIdade(idade);

cout << "Digite a altura da pessoa2 (em metros): ";
cin >> altura;
pessoa2.setAltura(altura);

cout << endl << "Dados da pessoa1:" << endl;
pessoa1.imprimirDados();

cout << endl << "Dados da pessoa2:" << endl;
pessoa2.imprimirDados();

return 0;
}

```

2. Crie uma classe denominada **Elevador** para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar: o andar atual (0 = térreo), o total de andares no prédio (excluindo o térreo), a capacidade do elevador e o número atual de pessoas no elevador.

A classe deve disponibilizar os seguintes métodos:

- **inicializa**: recebe como parâmetros a capacidade do elevador e o total de andares. O elevador começa sempre no térreo (andar 0) e vazio (0 pessoas);
  - **entra**: acrescenta uma pessoa ao elevador, se ainda houver espaço disponível;
  - **sai**: remove uma pessoa do elevador, se houver alguém dentro;
  - **sobe**: sobe um andar, desde que não esteja no último andar;
  - **desce**: desce um andar, desde que não esteja no térreo;
  - métodos **get...**: para obter os dados atuais do elevador.
3. Crie uma classe em C++ chamada **Relogio** para armazenar um horário, composto por hora, minuto e segundo. A classe deve conter:
    - um método **setHora** que receba hora, minuto e segundo como parâmetros;
    - um método **getHora** que retorne os valores atuais por referência;
    - um método que avance o horário em um segundo, ajustando os minutos e horas quando necessário.

4. Defina uma classe que represente um **círculo simples**, chamada **CirculoSimples**. A classe deve conter:

**Métodos privados** para:

- calcular a área do círculo;
- calcular a circunferência do círculo;
- calcular a distância entre os centros de dois círculos (recebendo outro objeto como parâmetro).

**Métodos públicos** para:

- definir o raio do círculo, dado um número real;
- aumentar o raio do círculo, dado um percentual de aumento;
- definir o centro do círculo, dada uma posição (X,Y);
- imprimir o valor do raio;
- imprimir o centro do círculo;
- imprimir a área do círculo.

Crie um programa principal para testar a classe. Observe que neste exercício usamos uma classe básica chamada `CirculoSimples`, que não utiliza herança.

5. Implemente uma classe `Televisao`. A televisão possui um controle de volume do som e um controle de canal. O controle de volume permite aumentar ou diminuir uma unidade por vez, dentro de limites (por exemplo, entre 0 e 100). O controle de canal permite:

- aumentar ou diminuir uma unidade no número do canal;
- trocar diretamente para um canal indicado;
- consultar o canal atual e o volume atual.

No programa principal, crie uma televisão, altere o canal algumas vezes, aumente o volume e exiba os valores atuais de canal e volume.

6. Implemente uma classe `CondicionadorAr`. O condicionador possui 10 níveis de potência (de 0 a 10). Cada nível reduz a temperatura em 1,8 °C. A variação máxima possível é de 18 °C (com potência 10). O condicionador nunca reduz a temperatura abaixo de 0 °C de variação.

A classe deve conter:

- um método para configurar a temperatura externa (ex: 25 °C);
- um método para configurar a potência (de 0 a 10);
- um método para calcular e retornar a temperatura ambiente resultante.

No programa principal, crie dois condicionadores. Defina temperaturas externas diferentes (ex: 25 °C e 31 °C). Ajuste o primeiro para potência 5 e o segundo para potência máxima (10). Exiba a temperatura ambiente resultante para cada um.

7. Implemente uma classe `Carro`. O tanque de combustível armazena até 50 litros de gasolina. O consumo é de 15 km por litro. A classe deve permitir:

- abastecer com uma certa quantidade de gasolina (sem ultrapassar a capacidade do tanque);
- mover o carro por uma determinada distância (em km), consumindo o combustível proporcional;
- retornar a quantidade de combustível restante e a distância total percorrida.

No programa principal:

- crie dois carros;
- abasteça o primeiro com 20 litros e o segundo com 30 litros;
- mova o primeiro por 200 km e o segundo por 400 km;
- exiba a distância percorrida e o combustível restante em cada carro.