

Lista de Exercícios - I

1. Escreva os métodos da seguinte classe:

```
class Complexo {  
    public:  
        Complexo(void);  
        Complexo(float re, float im);  
        float getReal(void);  
        float getImaginario(void);  
        void setReal(float re);  
        void setImaginario(float im);  
        void imprime(void);  
    private:  
        float real;  
        float imag;  
};
```

2. Crie uma classe para representar uma pessoa, com os atributos privados de nome, idade e id(rg). Crie os métodos públicos necessários para **setters** e **getters** e também um método para imprimir os dados de uma pessoa. Em seguida, escreva um programa main que faça uso da classe Pessoa, para isso crie duas instancias da classe pessoa, preencha as informações da primeira instancia do objeto pessoa utilizando os métodos **setters** e **getters** e as informações da segunda instância da classe Pessoa deve ser preenchida no momento da criação do objeto, ou seja, construa e utilize um método construtor para isso.
3. 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), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:
 - a) inicializa: que deve receber como parâmetros: a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazios);
 - b) entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
 - c) sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
 - d) sobe: para subir um andar (não deve subir se já estiver no último andar);
 - e) desce: para descer um andar (não deve descer se já estiver no térreo);
 - f) get e set.....: métodos para obter cada um dos os dados armazenados.

4. Implemente uma televisão. A televisão tem um controle de volume do som e um controle de seleção de canal. No programa principal, crie uma televisão e troque de canal algumas vezes. Aumente um pouco o volume, e exiba o valor de ambos os atributos. Para isso, considere as seguintes especificações:
 - a) O controle de volume permite aumentar ou diminuir a potência do volume de som em uma unidade de cada vez.
 - b) O controle de canal também permite aumentar e diminuir o número do canal em uma unidade, porém, também possibilita trocar para um canal indicado.
 - c) Também devem existir métodos para consultar o valor do volume de som e o canal selecionado.

5. Implemente um condicionador de ar. O condicionador possui 10 potências diferentes. Cada unidade da potência do condicionador diminui a temperatura do ambiente em 1.8oC. A variação que o condicionador consegue causar está no intervalo [0oC - 18oC], ou seja, zero graus de variação quando desligado e dezoito graus de variação quando ligado na potência máxima.
 - a) Através de um sensor, o condicionador é informado da temperatura externa. Dada essa temperatura e a potência selecionada, o condicionador calcula e retorna a temperatura do ambiente.
 - b) No programa principal, crie dois condicionadores. Informe duas temperaturas externas diferentes para cada um (ex: 25oC e 31o), ajuste o segundo em potência máxima (10) e o primeiro em potência média (5). Finalmente, exiba a temperatura resultante de cada ambiente.

6. Implemente um carro. O tanque de combustível do carro armazena no máximo 50 litros de gasolina. O carro consome 15 km/litro. No programa principal, crie 2 carros. Abasteça 20 litros no primeiro e 30 litros no segundo. Desloque o primeiro em 200 km e o segundo em 400 km. Exiba na tela a distância percorrida e o total de combustível restante para cada um. Para isto, na classe carro deve ser possível:
 - a) Abastecer o carro com uma certa quantidade de gasolina;
 - b) Mover o carro em uma determinada distância (medida em km);
 - c) Retornar a quantidade de combustível e a distância total percorrida.

7. Implemente em C++ as classes representadas nos diagramas abaixo e faça um programa principal para testar as classes abaixo.

Data
-dia: int -mes: int -ano: int
+initDt(int, int, int) : void +bissexto(void) : bool +imprimeData(void) : void

Retangulo
-lado1 : float -lado2 : float
+initRet(float, float) : void +area(void) : float +perimetro(void) : float

Referências

- Programação Orientada a Objetos em C++. Disponível em:
https://agostinhobritojr.github.io/tutorial/cpp/#_pre.prefacio
- Aula 2 – POO. Profa. Elaine Faria. Disponível em:
<http://www.facom.ufu.br/~elaine/disc/POO/Aula2-POO-Introducao.pdf>
- Introdução a Orientação a Objetos. Prof. Leonardo Gresta Paulino Murta. Disponível em: <http://www.ic.uff.br/~leomurta/courses/poo/aula7.pdf>
- Declaração de classes, métodos construtores. Prof. Bruno E. G. Gomes. Disponível em:
<https://docente.ifrn.edu.br/brunogurgel/disciplinas/2012/fprog/aulas/poo/POO-2-def-classes-construtores.pdf>
- Programação Orientada a Objetos. Prof. Marco Antonio M. Carvalho. Disponível em:
http://www.decom.ufop.br/marco/site_media/uploads/bcc221/03_aula_03.pdf
- Professor Paulo Villa, disponível em:
http://www.univasf.edu.br/~leonardo.campos/Arquivos/Disciplinas/POO_2007_2/POO_Lista_02.pdf
- Programação C/C++. Prof. Márcio Sarroglia Pinho, disponível em:
<https://www.inf.pucrs.br/~pinho/CPP/Exercicios/Classes/ExercCriarClasse1.html>