



1) A herança é um dos pilares da Programação Orientada a Objetos, que permite criar uma nova classe a partir de uma classe existente. Considerando o conceito de herança, avalie as afirmações abaixo.

I. Herança é uma propriedade que facilita a implementação de reuso;

II. Quando uma subclasse é criada, essa herda todas as características da superclasse, não podendo possuir propriedades e métodos próprios;

III. Herança múltipla é uma propriedade na qual uma superclasse possui diversas subclasse.

Quais dos itens acima são verdadeiros?

Apenas o item I é verdadeiro. A propriedade de herança permite que métodos ou atributos de classes pai sejam utilizados em classes filhas, facilitando implementação de lógicas semelhantes como no exemplo abaixo:

```
class Pessoa{
    public:
        void viver(){
            std::cout << "Eu acordo às 7h" << std::endl;
        };
};

class Estudante : public Pessoa{
    public:
        void viver(){
            Pessoa::viver();
            std::cout << "Eu procrastino até as 22h e durmo" << std::endl;
        };
};

int main(){
    Estudante a;
    a.viver(); //output : Eu acordo às 7h
              //          Eu procrastino até as 22h e durmo
}
```

Como visto no exemplo anterior, subclasses herdam características da superclasse, mas isso não impede de ter características próprias. anulando o item II.

Quanto a proposição III ser falsa, herança múltipla defini-se pela seguinte característica: uma subclasse herda características de diferentes classes, como no exemplo abaixo.

```
class A{};

class B{};

class X : public A, public B{}; /* Classe X herda todas as
                                características das classes A e B */
```



2) Desenvolva uma aplicação em C++, que seja possível utilizar herança múltipla, crie um arquivo de texto e explique o porque escolheu tal exemplo.

```
class VeiculoTerrestre{
public:
    int numPortas;
    int numRodas;
    int velTerra;
};

class VeiculoAquatico{
public:
    bool afunda;
    bool motorAvapor;
    int velAgua;
};

class veiculoAnfibio : public VeiculoTerrestre, public VeiculoAquatico{
public:
    std::string combuistivel;
    void transformar();
};

int main(){

    VeiculoTerrestre carro;
    VeiculoAquatico barco;
    veiculoAnfibio Schwimmwagen;

}
```

O exemplo anterior demonstra o uso de herança múltipla de maneira intuitiva. Veículos Anfíbios tratam-se de carros ou motos, geralmente militares, que funcionam tanto em terra quanto em água. Herdam portanto características de veículos terrestres (seu uso em terra, quantidade de rodas, etc.) e veículos aquáticos.

Dessa maneira, o veículo Schwimmwagen herda as características tanto de um carro por exemplo, como de um barco.