



Classe Carros

Keanu Sales / Matrícula 202104940028

START

MENU



Carros

- **horsepower: int**
- **upgradeLevel: int**

- **upgrade(int)**
- **printState()**

Essa classe possui 2 métodos e 2 atributos.

Atributos:

- **horsepower:**

Representa um número que mostra a potência dos carros, que são objetos.

- **upgradeLevel:**

Representa o nível de upgrade da parte mecânica dos carros, o que reflete diretamente em sua potência.



Carros

- **horsepower: int**
- **upgradeLevel: int**

- **upgrade(int)**
- **printState()**

Essa classe possui 2 métodos e 2 atributos.

Métodos:

- **upgrade(int):**

Atualiza o nível de upgrade dos carros, que vai de 0 a 5, o que impacta na potência do carro (objeto).

- **printState():**

Mostra o estado de potência e upgrade atuais do Carro.

Carros.h

```
#ifndef CARROS_H
#define CARROS_H

#include <iostream>           //Os includes são feitos aqui,
#include <string>             //no carros.cpp só é necessário
using namespace std;        //incluir o .h e no main só o carros.cpp

class Carros {
public:
    Carros();                //Constructor
    ~Carros();               //Destructor
    void upgrade(int);        //Método do upgrade, já descrito
    void printState(string) const; //Método do printState, já descrito

private:
    int horsepower, upgradeLevel; //Atributos
};

#endif //CARROS_H
```

Carros.cpp

```
#include "carros.h"

//Método que inicia o objeto com valores já definidos
Carros::Carros() {
    horsepower = 250, upgradeLevel = 0;
}

//Método que faz o upgrade, com valores difinidos
void Carros::upgrade(int level) {
    if (level == 0) this -> horsepower = 250;
    else if (level == 1) this -> horsepower = 400;
    else if (level == 2) this -> horsepower = 550;
    else if (level == 3) this -> horsepower = 700;
    else if (level == 4) this -> horsepower = 800;
    else this -> horsepower = 900;
    this -> upgradeLevel = level;
}

//Método que faz o print das estatísticas do objeto
void Carros::printStats(string nome) const {
    cout << "Horsepower do " << nome << ": " << horsepower << endl;
    cout << "Upgrade Level do " << nome << ": " << upgradeLevel << endl;
}
```

main.cpp

```
#include "carros.cpp"

int main() {
    int opcao, level, escolha;
    Carros lancerEvoX, audiTTRS; //Objetos
    system("clear||cls"); //Faz a limpeza do terminal
    while (1) { //Loop necessário para executar o programa enquanto o usuário não decide sair
        cout << "1 - Printar estado atual dos carros \n"
            << "2 - Fazer uma atualizacao em um dos carros \n"
            << "3 - Sair do programa \n" //Para o loop e sai do programa
            << "Qual opcao voce escolhe? " ;
        cin >> opcao;
        switch (opcao) {
            case 1: { //Há código aqui} //Os cases serão mostrados nos próximos slides
            case 2: { //Há código aqui}
            case 3: { //Há código aqui}
            default: { //Há código aqui}
        }
    }
}
```

main.cpp

```
case 1:
    system("clear||cls");
    lancerEvoX.printState("Lancer");           //Printa o estado do 1º objeto
    audiTTRS.printState("TTRS");               //Printa o estado do 2º Objeto
    break;
case 2:                                         //Será mostrado no próximo slide, pois é muito grande
case 3:
    system("clear||cls");
    cout << "Ate a proxima vez!\n";
    exit(0);                                   //Para aquele while do slide anterior
default:
    cin.clear(); cin.ignore();
    system("clear||cls");
    cout << "Valor Invalido! Tente de novo.\n"; //Caso o usuário não entre com alguma opção
    break;
```

main.cpp

case 2:

```
system("clear||cls");
cout << "Digite o nivel da att [0, 5]: ";           //O usuário deve escolher a atualização
cin >> level;
if (0 <= level && level <= 5) {                     //Caso o usuário digite um valor dentro do range
    cout << "1: Lancer, 2: TTRS - Escolha: ";       //O usuário deve escolher qual objeto
    cin >> escolha;
    switch (escolha) { //Há código aqui              //Será mostrado no próximo slide
} else {                                             //Caso contrário
    cin.clear(); cin.ignore();
    system("clear||cls");
    cout << "Digite um valor no intervalo dado!\n";
}
break;
```


main.cpp

```
switch (escolha) {
    case 1:
        lancerEvoX.upgrade(level);           //A atualização é aplicada no 1° objeto
        system("clear||cls");
        cout << "Atualizacao realizada com sucesso!\n";
        break;
    case 2:
        audiTTRS.upgrade(level);             //A atualização é aplicada no 2° objeto
        system("clear||cls");
        cout << "Atualizacao realizada com sucesso!\n";
        break;
    default:
        system("clear||cls");                 //Caso contrário
        cout << "Escolha um dos carros mostrados!\n";
        break;
}
```