



Elementos estáticos

- Atributos e métodos que não dependem de valores específicos baseados na instância do objeto em questão
- Elementos que dependem da classe, e não dos objetos criados
- Elementos compartilhados entre as instâncias de uma classe
- Operador **static**

Como funcionam

```
class Banco {
public:
    int codigoDoBanco;
    static double cotDolar = 1.81;

    Banco::Banco(int codigoDoBanco) {
        this->codigoDoBanco = codigoDoBanco;
    }

    static double Banco::conversaoDeMoeda(double reais) {
        return reais / Banco::cotDolar;
    }
}

double Banco::cotDolar = 1.81;
```

```
cout << Banco::cotDolar; // 1.81
cout << Banco::conversaoDeMoeda(100); // 55.24
Banco::cotDolar = 1.83;
cout << Banco::cotDolar; // 1.83
Banco *b1 = new Banco(2);
Banco *b2 = new Banco(3);
cout << b1->conversaoDeMoeda(100); // 54.64
```

Memória

Classes estáticas



- Existem classes estáticas?
- O operador **static** não pode ser utilizado para definir uma classe estática
- Classes auxiliares
- Classes podem ser estáticas se definirem um método virtual puro

```
class Geometria {
public:
    static double pi;

    static double Geometria::calculaArea(double raio) {
        return (raio * raio) * Geometria::pi;
    }
}

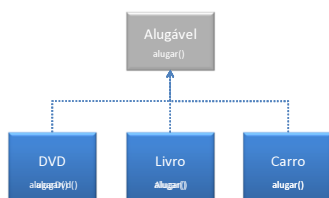
double Geometria::pi = 3.14;
```

Interfaces




- Protocolo que define a assinatura de um método que virá a ser implementado pelas classes que façam uso desta interface
- <tipoDeRetorno> <nome> (parâmetros);
- float** calculaArea(**int** largura);
- Contrato assinado por uma classe se comprometendo a implementar o método da interface

Exemplo de interface



Interfaces na prática



- Criando uma interface: **virtual ... = 0**


```
class Alugavel {
public:
    virtual void alugar() = 0;
    virtual void retornarLocacao() = 0;
}
```

- Implementando uma interface

```
class Livro : public Alugavel {
public:
    virtual void alugar();
    virtual void retornarLocacao();
}
```


```
#include "Livro.h"
void Livro::alugar() {
    // implementação
}
void Livro::retornarLocacao() {
    // implementação
}
```

Interfaces



- Geralmente nomes que indicam uma qualidade
 - *Alugavel, Dirigivel, Deletavel*
- É possível incluir atributos em interfaces
- Uma classe pode implementar uma ou mais interfaces utilizando herança múltipla
- Métodos construtores e destrutores das "interfaces"

Abstrações



- Virtualização de objetos-conceitos
 - Exemplos: veículo, cor, móvel, ser vivo
- Abstração de um conceito que as subclasses irão herdar
- Padronização: evita que cada subclasse crie sua própria assinatura do método
- Facilita a utilização em laços de repetição e navegação de listas de objetos
- É uma classe não instanciável

Abstrações



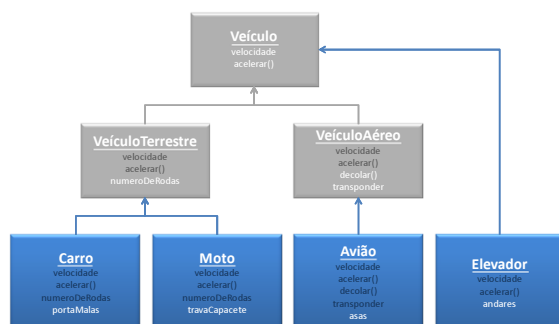
- Somente classes abstratas podem conter métodos abstratos
- Classes abstratas podem conter atributos e métodos não abstratos
- A primeira classe não-abstrata deve implementar as assinaturas dos métodos abstratos da classe de qual deriva
- Classes abstratas podem herdar de outras classes abstratas sem implementar os métodos

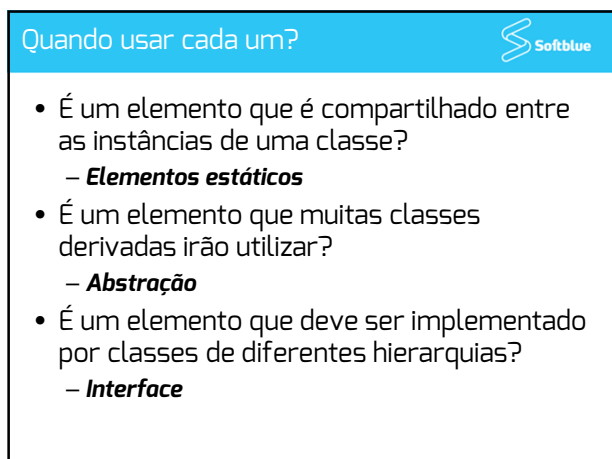
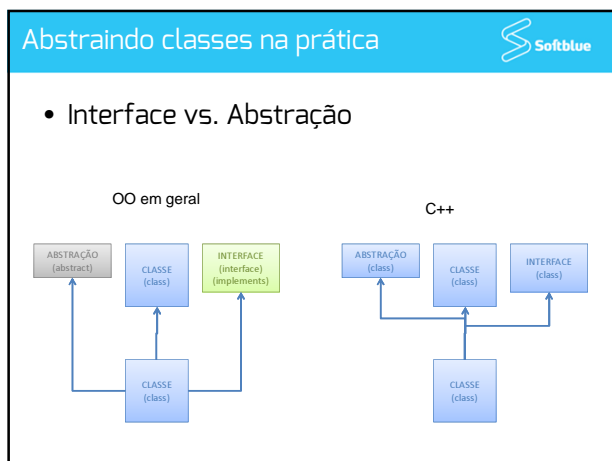
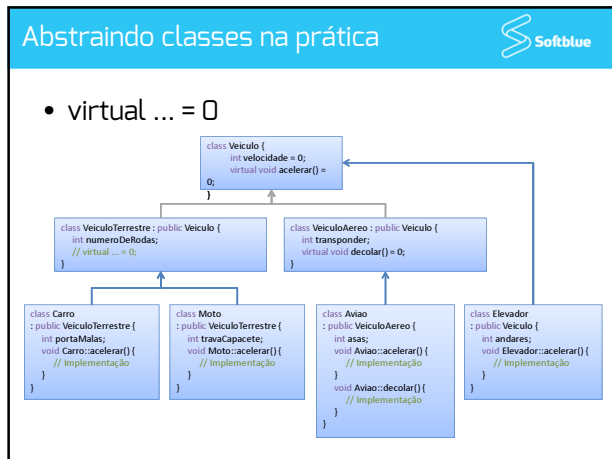
Abstrações




- Classes abstratas podem contratar interfaces, e podem ou não implementar os métodos
- A primeira classe não abstrata de uma hierarquia deve implementar ou herdar implementados todos os métodos abstratos
- Entenderemos melhor sobre qual a primeira classe é obrigada a implementar os métodos nos próximos slides

Exemplos de conceitos



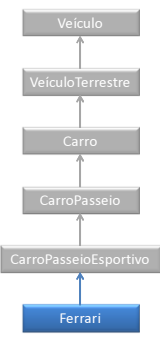


Considerações sobre OO



- Benefícios
- É uma tendência
- Uso no mercado cada vez maior
- Abstrair a realidade até que ponto para meu projeto?
- Coesão

Abstração da realidade deve ser feita levando em conta as características de cada projeto, abstraindo somente os dados necessários.



```
graph BT; Ferrari[Ferrari] --> CarroPasseioEsportivo[CarroPasseioEsportivo]; CarroPasseioEsportivo --> CarroPasseio[CarroPasseio]; CarroPasseio --> Carro[Carro]; Carro --> VeiculoTerrestre[VeiculoTerrestre]; VeiculoTerrestre --> Veiculo[Veiculo];
```

Aulas práticas e manuais on-line



Assista agora as aulas práticas.

[Clique aqui](#) para visualizar as aulas práticas disponíveis
