



AED - Algoritmos e Estruturas de Dados

Aula 5

Prof. Rodrigo Mafort

Métodos virtuais

- Na última aula conversamos sobre a herança.
- Vamos considerar o seguinte modelo:
 - Vamos implementar polígonos: quadrados, círculos e triângulos.
 - Para isso, vamos construir uma classe chamada de Polígono
 - Essa classe vai ser ancestral de todos os nossos polígonos
- Vamos considerar que todas as classes tem um método chamado Desenhar().
- Como esse método é comum a todos, vamos definir esse método na classe ancestral.

Métodos virtuais

- Quando um método é marcado como virtual estamos definindo que ele pode ser sobrescrito pela classe filha.
- No exemplo, polígono tem um método Desenhar()
- Só que não sabemos desenhar polígonos quaisquer.
- Mas ao definir as classes filhas de polígono, podemos sobrescrever esse método.
- Entretanto: todo polígono terá um método Desenhar()
- Isso se comporta como um contrato firmado durante a herança

Métodos e classes abstratas

- Quando definimos uma classe com métodos virtuais, muitas vezes não queremos que essa classe seja realmente instanciada (não queremos objetos dessa classe).
- Nesse caso, essa classe assume ainda mais esse aspecto de contrato: toda classe que herdar deverá cumprir o contrato, sobrescrevendo o métodos marcados como virtuais.
- Para indicar que um método é abstrato precisamos:
 1. Marcar como virtual
 2. Indicar no arquivo.h que esse método recebe 0.
- `virtual void Desenhar() = 0;`
- Importante: a existência de um método abstrato implica que toda a classe é abstrata.
- Desta forma, essa classe não poderá ser instanciada.

Observe o exemplo:

```
int main()
{
    Quadrado q;
    Circulo c;

    vector<Poligono*> vet;
    vet.push_back(&q);
    vet.push_back(&c);

    for (int i = 0; i < vet.size(); i++)
        vet[i]->Desenhar(); //O que será impresso?

    return 0;
}
```

Polimorfismo

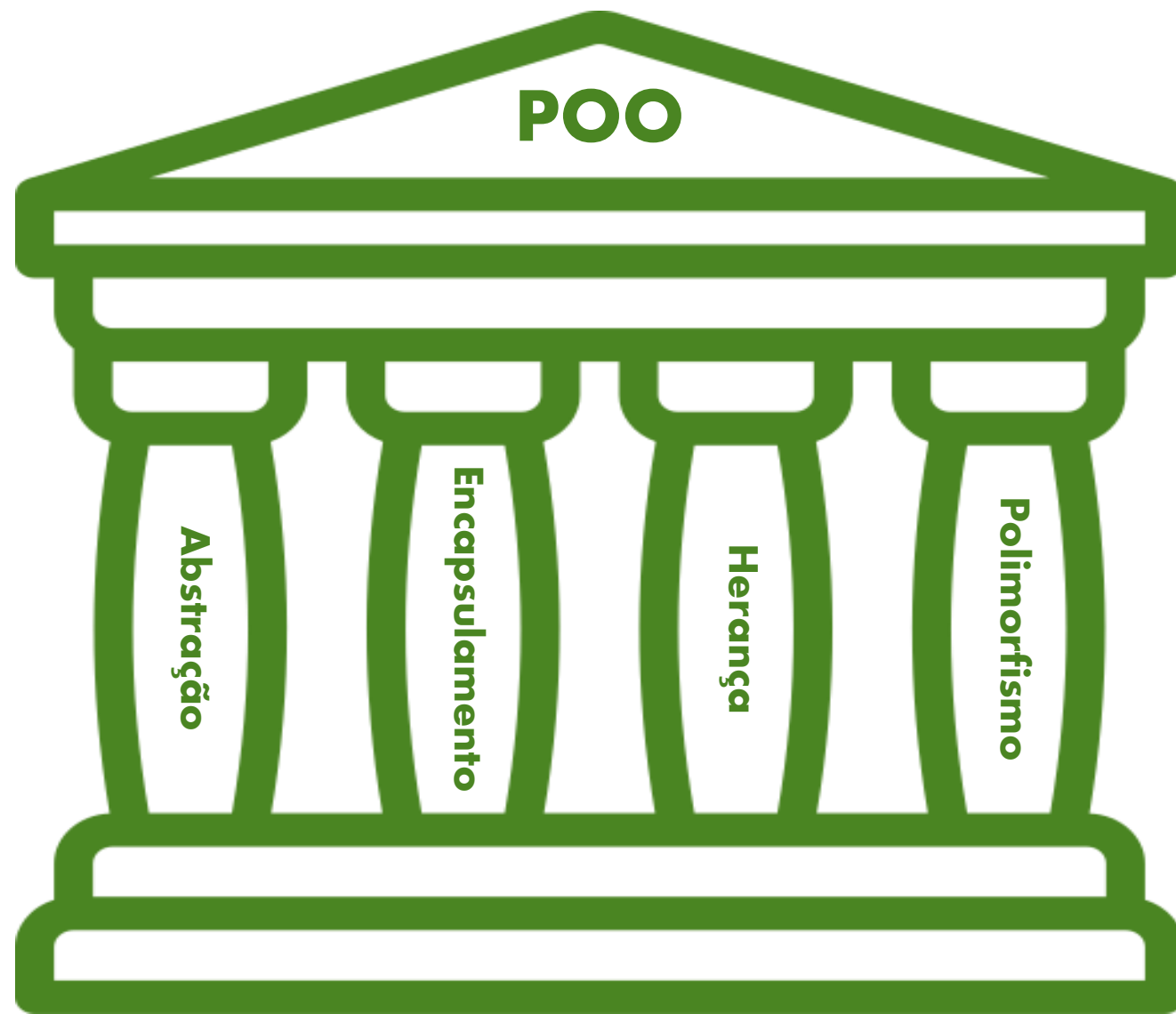
- Poli = muitas
- Morphos = formas
- Polimorfismo = muitas formas
- Mas o que vai adotar muitas formas???
 - Já vimos vários exemplos de polimorfismo ao longo do curso

Polimorfismo

- Temos dois casos de polimorfismo:
 1. Funções com mesmo nome, mas com parâmetros diferentes
 - Chamado também de Sobrecarga de Métodos (Overload)
 - Usamos esse conceito em diversos exemplos quando definimos dois construtores: um com parâmetros e o outro sem.
 2. Métodos virtuais através da herança
 - Na última aula, definimos três classes: Polígono, Circulo e Quadrado
 - Todas as classes apresentavam um método Desenhar
 - No exemplo, observamos que apesar de tratar todos como Polígonos, chamamos o método apropriado de cada objeto (Circulo e Quadrado)
 - Logo, apesar de Círculo e Quadrado terem se passado por Polígonos, ainda podemos acessar seus métodos especializados.

Conceitos POO

- Desta forma, observamos os quatro pilares da orientação a objetos:
 - Abstração
 - Encapsulamento
 - Herança
 - Polimorfismo



Exercício

- Continue o exercício anterior dos Polígonos e:
 - Crie a classe Polígono que contém métodos virtuais para calcular área e perímetro.
 - Depois, crie as classes Círculo, Retângulo e Quadrado, que herdam de Polígono e escreva implementações para os métodos de cálculo de cada subclasse.
 - Implemente construtores para:
 - Inicializar a classe partindo de suas dimensões
 - Inicializar a classe partindo de coordenadas (por exemplo, centro do círculo e raio ou dois vértices do retângulo).
 - Em seguida, crie um vetor com diferentes polígonos e apresente as áreas e os perímetros de cada um.