Fatec
Rio Claro

Linguagem de Programação II

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Pilares do Paradigma

Fatec
Rio Claro

O paradigma é sustentado por quatro pilares principais:

Abstração: focar nas características essenciais do objeto, ignorando detalhes irrelevantes.

Encapsulamento: proteger os dados, controlando como são acessados ou modificados.

Herança: permitir que uma classe herde atributos e métodos de outra, facilitando o reuso de código.

Polimorfismo: possibilitar que diferentes objetos respondam de formas distintas a uma mesma mensagem/método.

Herança

Herança

Fatec
Rio Claro

Herança em POO é a capacidade de criar uma classe que se baseia (estende) em uma classe existente. Ao criar programas grandes, você frequentemente usará classes que fornecem recursos gerais muito úteis.

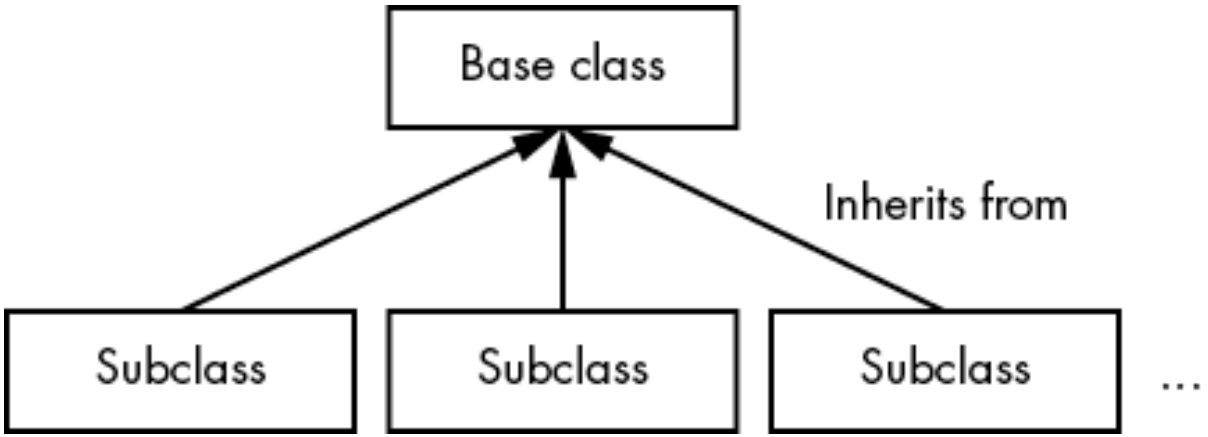
Caso deseje construir uma classe semelhante a uma classe já existente, mas que faça algumas coisas de forma.

Com herança, falamos sobre o relacionamento entre duas classes, normalmente chamadas de classe ***base*** e ***subclasse***.

Várias Classes Herdando da Mesma Classe Base

Fatec Rio Claro

Várias classes diferentes podem herdar da mesma classe base. Você pode construir uma classe base bem geral e, em seguida, construir qualquer número de subclasses que herdem dela.



Várias Classes Herdando da Mesma Classe Base

Fatec Rio Claro

Cada uma das diferentes subclasses pode ser uma variante (uma versão mais específica) da classe base genérica.

Cada subclasse pode sobrescrever quaisquer métodos da classe base que desejar ou precisar, independentemente de qualquer outra subclasse.

Prática

Fatec
Rio Claro

Analisar o código

veiculos.py

Classe Abstrata

Fatec
Rio Claro

Uma classe que não se destina a ser instanciada diretamente, mas apenas a ser usada como classe base por uma ou mais subclasses.

Em algumas outras linguagens, uma classe abstrata é chamada de classe virtual.

Método Abstrato

Fatec
Rio Claro

Quando utilizamos método abstrato, um método que deve ser sobrescrito em todas as subclasses.

Frequentemente, uma classe base não consegue implementar corretamente um método abstrato porque não conhece os dados detalhados com os quais deve operar, ou porque pode não ser possível implementar um algoritmo geral.

Em vez disso, todas as subclasses precisam implementar sua própria versão do método abstrato.

Método Abstrato

Fatec
Rio Claro

O Python não possui uma palavra-chave para designar uma classe ou método como abstrato.

No entanto, a Biblioteca Padrão do Python contém o módulo **abc**, abreviação de *abstract base class* (classe base abstrata), que foi projetado para ajudar desenvolvedores a criar classes e métodos base abstratos.

A Dificuldade de Programar com Herança

Fatec
Rio Claro

Ao desenvolver usando herança, pode ser difícil entender o que colocar e onde. Você está constantemente se perguntando:

Esta variável de instância deve estar na classe base?

Existe código comum suficiente nas subclasses para criar um método na classe base?

Quais são os parâmetros apropriados para um método em uma subclasse?

Quais são os parâmetros e padrões apropriados para serem usados em uma classe base que espera ser sobrescrita ou chamada de uma subclasse?

A Dificuldade de Programar com Herança

Fatec Rio Claro

Tentar entender as interações entre todas as variáveis e métodos em uma hierarquia de classes pode ser uma tarefa extremamente difícil, complexa e frustrante.

Isso é especialmente verdadeiro ao ler o código de uma hierarquia de classes desenvolvida por outro programador.

Para entender completamente o que está acontecendo, muitas vezes você precisa se familiarizar com o código nas classes base em toda a hierarquia.

Prática

Fatec
Rio Claro

Analisar o código

animais.py

Polimorfismo

Polimorfismo

Fatec
Rio Claro

Programadores frequentemente usam o termo "enviar uma mensagem" quando falamos sobre código cliente chamando um método de um objeto.

O que o objeto deve fazer ao receber a mensagem depende do objeto.

Com o polimorfismo, podemos enviar a mesma mensagem para vários objetos, e cada objeto reagirá de forma diferente dependendo do que foi projetado para fazer e dos dados disponíveis.

Enviando Mensagens para Objetos do Mundo Real

Fatec Rio Claro

Vejamos o polimorfismo no mundo real, usando carros como exemplo.

Todos os carros têm um pedal de acelerador. Quando o motorista pressiona esse pedal, ele envia a mensagem de "acelerar" para o carro.

O carro que ele dirige pode ter um motor de combustão interna, um motor elétrico ou ser um híbrido.

Cada um desses tipos de carro tem sua própria implementação do que acontece quando recebe a mensagem de aceleração, e cada um se comporta de acordo.

Enviando Mensagens para Objetos do Mundo Real

Fatec
Rio Claro

O polimorfismo facilita a adoção de novas tecnologias.

Se alguém desenvolvesse um carro movido a energia nuclear, a interface do usuário do carro permaneceria a mesma — o motorista ainda pressionaria o pedal do acelerador para enviar a mesma mensagem —, mas um mecanismo muito diferente faria o carro movido a energia nuclear ir mais rápido.

Enviando Mensagens para Objetos do Mundo Real

Fatec
Rio Claro

Como outro exemplo do mundo real, imagine que você entra em uma sala grande com um conjunto de interruptores que controlam uma variedade de luzes diferentes.

Algumas das lâmpadas são incandescentes antigas, algumas são fluorescentes e algumas são lâmpadas LED mais novas.

Ao girar todos os interruptores para cima, você envia a mensagem de "ligar" para todas as lâmpadas.

Os mecanismos subjacentes que fazem com que as lâmpadas incandescentes, fluorescentes e de LED emitam luz são completamente diferentes, mas cada um atinge o objetivo pretendido pelo usuário.

Polimorfismo em Programação

Fatec
Rio Claro

Polimorfismo se refere à maneira como o código cliente pode chamar um método com exatamente o mesmo nome em diferentes objetos, e cada objeto fará o que for necessário para implementar o significado desse método para aquele objeto.

O exemplo clássico de polimorfismo é considerar um código que representa diferentes tipos de animais de estimação. Digamos que você tenha uma coleção de cães, gatos e pássaros, e cada um deles entenda alguns comandos básicos. Se você pedir a esses animais para falarem (ou seja, enviar a mensagem "falar" para cada um), os cães dirão "latir", os gatos dirão "miau" e os pássaros dirão "piu".

Prática

Fatec
Rio Claro

Analisar o código

PetsPolymorphism.py

Polimorfismo em Programação

Fatec
Rio Claro

Cada classe possui um método speak(), mas o conteúdo de cada método é diferente.

Cada classe faz o que precisa fazer em sua versão deste método; o nome do método é o mesmo, mas tem implementações diferentes.

Fatec
Rio Claro

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Prática

Fatec
Rio Claro

No Pandas, existem métodos de leitura:

pd.read_csv("dados.csv")

pd.read_excel("dados.xlsx")

pd.read_json("dados.json")

pd.read_sql(query, conexao)

pd.read_html("pagina.html")

Apesar dos nomes variarem um pouco (read_csv, read_excel, etc.), todos compartilham a mesma ideia polimórfica: ler dados de uma fonte qualquer e transformá-los em um **DataFrame**.

Ou seja, não importa se o dado vem de um CSV, de um Excel, de um banco SQL ou de uma página HTML — o método específico sabe como interpretar aquela fonte, mas o resultado final é sempre do mesmo tipo (DataFrame).

Prática

Fatec
Rio Claro

Elabore um exemplo que faça uso de polimorfismo e herança. Pesquise e faça uso de classes abstratas e métodos abstratos.

Utilize o contexto que achar ideal para o seu exemplo didático.

Bibliografia

KALB, Irv. Object-Oriented Python. San Francisco: No Starch Press, 2022. 416 p.