**Nome:** Layslla Eduarda Oreti **1DS-TB (18)**

**Avaliação Orientação a Objeto**

**Herança**

**O que é?**

Sendo considerada um dos pilares em Programação Orientada a Objetos (POO), a herança é um mecanismo que possibilita que uma classe (chamada filha) herde atributos e métodos de outra classe (chamada classe pai). Isso permite a reutilização de código e a criação de uma estrutura de classes com hierarquia.

**Classe pai ou superclasse:** Classe que concede as características.

**Classe filho ou subclasse:** Classe que recebe as características da classe pai/superclasse.

**Exemplo Prático**

Enviado em um arquivo a parte.

**Benefícios na arquitetura de sistemas orientados a objetos**

1. **Reutilização de código:** Ao herdar de uma classe existente, é possível aproveitar os métodos e atributos já implementados anteriormente, o que evita reescrever o mesmo código várias vezes, o que reduz a duplicação de código e também facilita a manutenção.
2. **Extensibilidade:** A herança permite a criação de classes especializadas a partir de uma classe base, adicionando funcionalidades específicas, o que possibilita a estender o comportamento das classes existentes, sem a necessidade de modificar o código original.
3. **Organização e estrutura do código:** Ao utilizar a herança, é possível criar hierarquias de classes, que refletem a estrutura do problema que está sendo resolvido, o que torna o código mais modular, organizado e de fácil compreensão.

**Polimorfismo**

**O que é?**

Em Programação Orientada a Objetos (POO), polimorfismo é a capacidade que uma função possui para tratar objetos de diferentes classes de forma uniforme, o que é possível pois as diferentes classes podem ter métodos com o mesmo nome, mas que possuem implementações distintas. Em geral, o polimorfismo é a capacidade que os objetos de classes diferentes possuem de responder a mesma mensagem (método) de diferentes jeitos.

**Benefícios na arquitetura de sistemas orientados a objetos**

1. **Flexibilidade no uso de objetos:** O mesmo método pode ser operado em objetos de classes diferentes, desde que compartilhem a mesma superclasse ou interface.
2. **Facilita a implementação de padrões de projeto:** Muitos padrões se apoiam fortemente com o uso do polimorfismo.
3. **Substituição de comportamentos no tempo de execução:** É ideal para sistemas dinâmicos no qual os comportamentos têm que ser definidos no tempo de execução.

**Diferenças e Relações**

A principal diferença entre Herança e Polimorfismo é o seu objetivo dentro da Programação Orientada a Objetos (POO). A herança permite a criação de novas classes a partir de outras existentes, criando uma reutilização de código, não havendo necessidade de reescrever sempre quando for necessário, pois ele cria a estrutura que o polimorfismo utilizará para permitir que os mesmos métodos respondam de maneiras distintas.

A herança define o que cada classe irá herdar, já o polimorfismo como essas classes irão se comportar.

**Referências bibliográficas**

<https://awari.com.br/heranca-em-python-aprenda-a-utilizar-esse-recurso-poderoso/#:~:text=hierarquias%20de%20classes.-,Exemplo%20de%20heran%C3%A7a%20em%20Python,marca%20%3D%20marca%20self>.

<https://www.datacamp.com/pt/tutorial/python-inheritance>

<https://www.alura.com.br/apostila-python-orientacao-a-objetos/heranca-e-classes-abstratas?srsltid=AfmBOoqXBjeWC7P3HCjlhtyEYc5oyOIY0mfSUjT0_jl_Xy2Ezm10wKd5>

<https://awari.com.br/aprenda-sobre-heranca-e-polimorfismo-em-python-conceitos-essenciais-para-programadores-iniciantes-e-avancados/>