Aluno: Victor do Espirito Santo Farias

Matricula: 202403557908

**Trabalho de Paradigmas de Programação em Python**

Quais são os principais paradigmas de programação e quais linguagens implementam cada um deles? Geração de linguagens.

Programação Imperativa (C, Java, Python e Ruby),

Orientada a Objetos (Java, C++, C#, Python e Ruby),

Funcional (Haskell, Scala, Lisp, Clojure e Erlang),

Lógica (Prolog e Mercury),

Declarativa (SQL, HTML e CSS),

Concorrente (Java com threads, Go, Erlang e Rust),

Reativa (JavaScript com RxJS, Scala com Akka e Kotlin)

**Gerações de linguagem:**

1ª geração: Linguagem de máquina (códigos binários)

2ª geração: Linguagem de montagem (assembly)

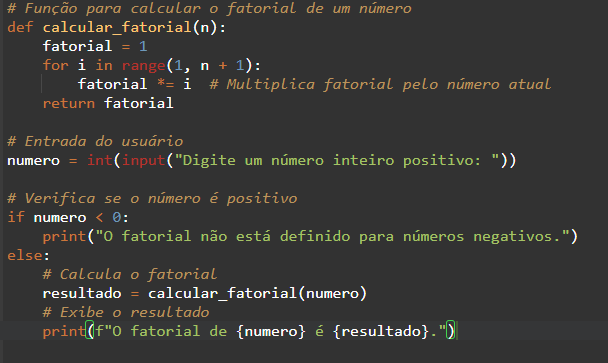
3ª geração: Linguagens de alto nível (C, Java, Python)

4ª geração: Linguagens declarativas e de domínio específico (SQL)

5ª geração: Linguagens para inteligência artificial e programação lógica (Prolog, Lisp)

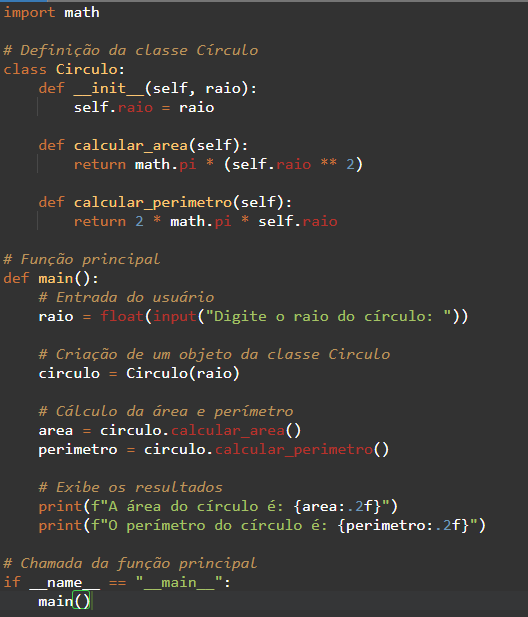
Exemplos de cada paradigma de Python?

Programação Imperativa:



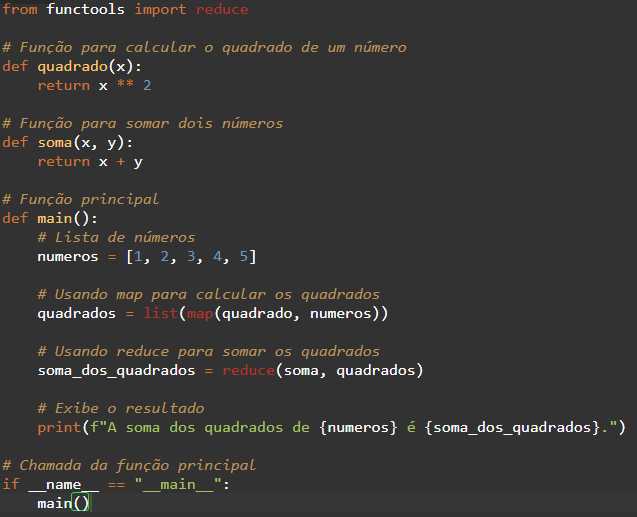
Este exemplo ilustra a abordagem imperativa, onde as instruções são executadas em sequência e a lógica é explicitamente definida passo a passo.

Orientada a Objetos (POO):



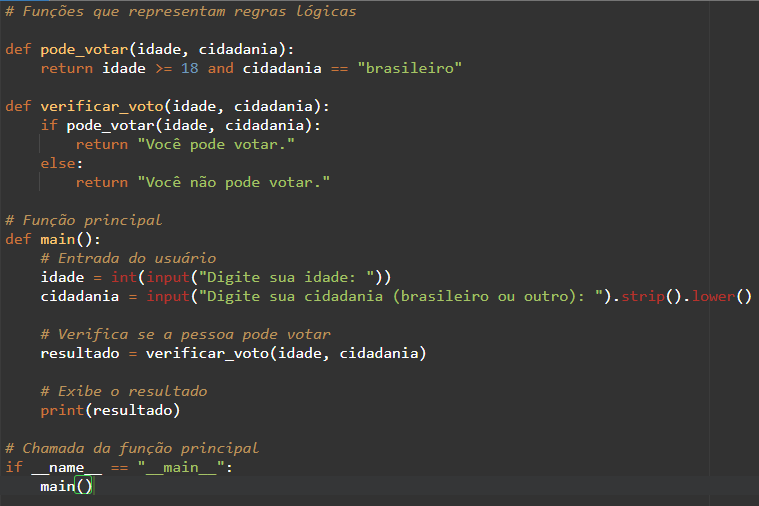
Esse exemplo demonstra a POO, onde a lógica é encapsulada em classes e métodos, promovendo a reutilização e organização do código.

Programação Funcional:

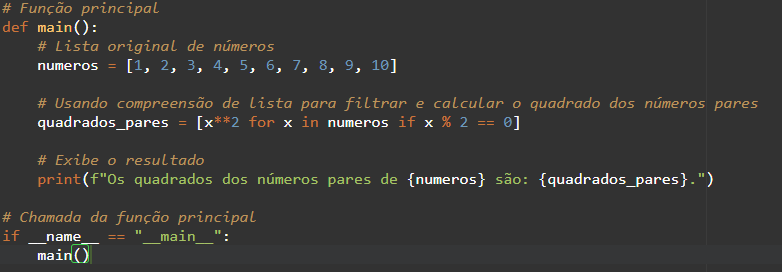


Esse exemplo ilustra a programação funcional, utilizando funções de ordem superior e evitando estados mutáveis, promovendo uma abordagem declarativa.

Programação Lógica:

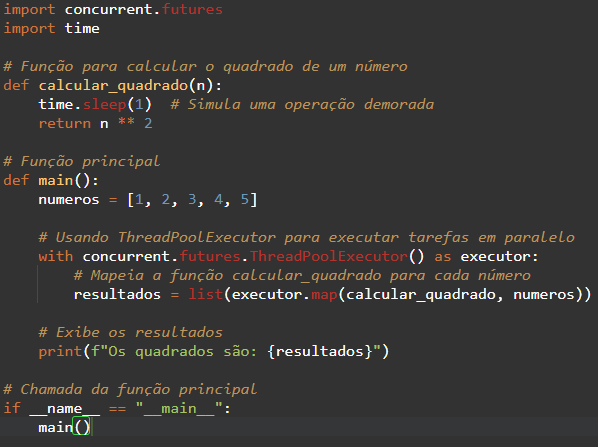


Embora esse exemplo não utilize um sistema lógico complexo como Prolog, ele ilustra a ideia de aplicar regras lógicas em Python.

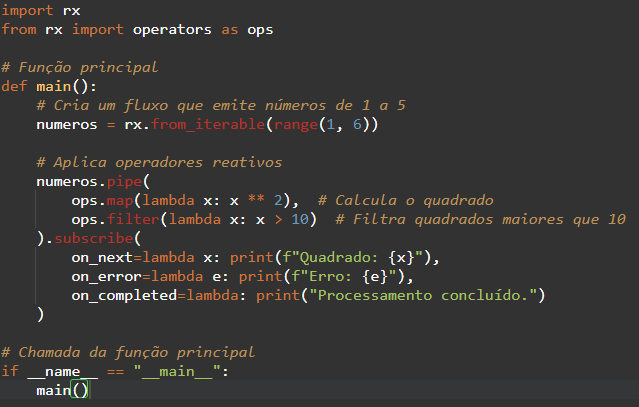
Programação Declarativa:

Esse exemplo ilustra a programação declarativa ao expressar o que queremos alcançar de maneira clara e concisa, evitando uma abordagem mais imperativa que envolveria loops explícitos.

Programação Concorrente:



Programação Reativa:



POO em Python e aplicação dos 4 pilares em POO

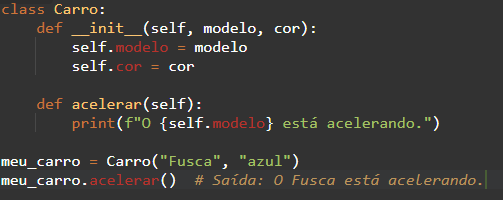
A Programação Orientada a Objetos (POO) em Python é um paradigma de programação que organiza o código em torno de "objetos", que são instâncias de classes. A POO permite a modelagem de entidades do mundo real de forma mais intuitiva e modular. Aqui estão os conceitos principais da POO em Python:

**Classes e Objetos:**

Classe: É uma estrutura que define um tipo de objeto, incluindo suas propriedades (atributos) e comportamentos (métodos).

Objeto: É uma instância de uma classe. Cada objeto pode ter seus próprios valores para os atributos definidos na classe.

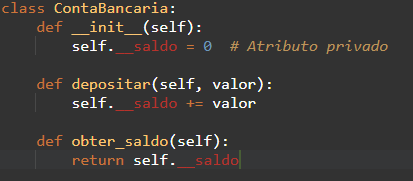
Exemplo:



**Encapsulamento:**

O encapsulamento é o conceito de esconder os detalhes internos de um objeto e expor apenas o que é necessário. Em Python, isso é feito através de atributos e métodos privados e públicos.

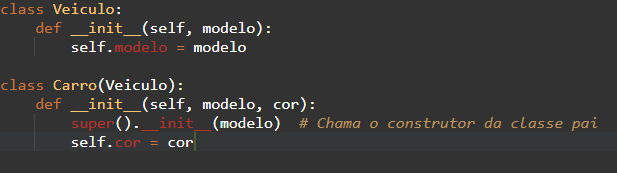
Exemplo:



**Herança:**

A herança permite que uma classe herde atributos e métodos de outra classe. Isso promove a reutilização do código e a criação de hierarquias de classes.

Exemplo:



**Polimorfismo:**

O polimorfismo permite que métodos com o mesmo nome se comportem de maneira diferente em classes diferentes. Isso pode ser feito através de sobrecarga de métodos ou através de métodos que são redefinidos nas subclasses.

Exemplo:

