

# **Explorando a Tecnologia Python: Projeto de uma Calculadora de Área e Perímetro com Orientação a Objetos**

Nome: Eron Teixeira de Moraes

Faculdade Ampli

Matéria: Linguagem de programação

## Sumário

1. Introdução:.....	3
2. História do Python:.....	4
3. Desenvolvimento da Linguagem Python:.....	5
4. Projeto: Calculadora de Área e Perímetro: .....	6
4.1. Código do Programa Modelo .....	6
5. Conclusão: .....	8
6. Bibliografia: .....	9

## 1. Introdução:

A linguagem de programação Python tem se destacado cada vez mais no mundo da programação devido às suas características como simplicidade, legibilidade e versatilidade. Neste trabalho, iremos explorar o universo do Python, abordando sua história, desenvolvimento e aplicação prática por meio de um projeto de uma calculadora de área e perímetro utilizando orientação a objetos.

Python foi criado por Guido van Rossum e lançado pela primeira vez em 1991. Sua filosofia de design enfatiza a legibilidade do código, encorajando uma sintaxe clara e organizada. A linguagem foi desenvolvida com o objetivo de ser fácil de aprender e usar, tornando-a ideal tanto para iniciantes quanto para programadores experientes.

Uma das principais vantagens do Python é sua vasta biblioteca padrão, que oferece uma ampla gama de módulos e funcionalidades prontos para uso. Além disso, a comunidade Python é extremamente ativa, o que significa que você pode encontrar facilmente suporte e recursos adicionais online.

Para exemplificar a aplicação prática do Python, vamos criar um projeto de uma calculadora de área e perímetro. Utilizaremos conceitos de orientação a objetos para criar classes que representam diferentes formas geométricas, como retângulos e círculos.

Primeiramente, criaremos a classe "Retângulo" com os atributos comprimento e largura. Em seguida, implementaremos métodos para calcular a área e o perímetro do retângulo. Faremos o mesmo processo para a classe "Círculo", utilizando o atributo raio e os métodos para calcular a área e o perímetro do círculo.

Após a definição das classes e métodos, poderemos criar instâncias dessas classes e utilizar seus métodos para calcular a área e o perímetro das formas geométricas desejadas.

Através desse projeto, poderemos explorar os conceitos fundamentais da orientação a objetos, como encapsulamento, herança e polimorfismo, enquanto aplicamos as funcionalidades do Python para realizar cálculos precisos de área e perímetro.

Em resumo, a linguagem de programação Python tem conquistado destaque devido à sua simplicidade, legibilidade e versatilidade. Neste trabalho, abordamos sua história, desenvolvimento e aplicação prática por meio de um projeto de uma calculadora de área e perímetro com orientação a objetos. Essa é apenas uma das muitas aplicações possíveis com Python, e sua flexibilidade permite que seja utilizada em uma ampla gama de áreas, desde desenvolvimento web até análise de dados.

## 2. História do Python:

Python foi criada no final dos anos 1980 por Guido van Rossum, um programador holandês. Ele desenvolveu a linguagem com o objetivo de criar uma linguagem fácil de ler, com uma sintaxe clara e uma abordagem pragmática. Python foi inspirada em outras linguagens, como ABC, Modula-3 e C, incorporando as melhores características dessas linguagens.

Uma das principais filosofias por trás do Python é o princípio do "Zen do Python", que enfatiza a legibilidade e a simplicidade do código. Esse princípio é representado pelo famoso "PEP 20" (Python Enhancement Proposal 20), que inclui afirmações como "Legibilidade conta" e "Simples é melhor do que complexo".

Python ganhou popularidade rapidamente devido à sua sintaxe intuitiva e facilidade de uso. A linguagem permite que os programadores expressem conceitos complexos de forma concisa e elegante. Além disso, a comunidade Python é conhecida por sua abertura e colaboração, o que contribuiu para o desenvolvimento de uma vasta biblioteca de módulos e pacotes que tornam o Python extremamente versátil.

Uma das características distintivas do Python é o uso de indentação significativa. Em vez de utilizar chaves ou palavras-chave para delimitar blocos de código, Python utiliza a indentação para indicar a estrutura do programa. Isso resulta em um código mais limpo e legível, incentivando boas práticas de programação.

A versatilidade do Python é evidente em suas diversas aplicações. A linguagem é amplamente utilizada no desenvolvimento web, seja na criação de sites ou no desenvolvimento de aplicativos web mais complexos. Também é uma escolha popular para automação de tarefas, scripting, análise de dados, inteligência artificial e aprendizado de máquina.

Além disso, o Python possui uma comunidade ativa e solidária, com uma ampla gama de recursos disponíveis para ajudar os programadores a aprender e aprimorar suas habilidades. Existem inúmeros tutoriais, documentação detalhada, fóruns de discussão e grupos de usuários que podem auxiliar na resolução de problemas e no compartilhamento de conhecimentos.

Em resumo, Python é uma linguagem de programação poderosa, versátil e de fácil leitura, que foi desenvolvida por Guido van Rossum no final dos anos 1980. Sua sintaxe clara, legibilidade e vasta biblioteca de módulos contribuíram para sua popularidade em diferentes áreas, desde desenvolvimento web até análise de dados e aprendizado de máquina. Com uma comunidade ativa e recursos abundantes, Python continua a evoluir e se destacar no mundo da programação.

### **3. Desenvolvimento da Linguagem Python:**

Desde sua criação, Python tem passado por um processo contínuo de desenvolvimento e aprimoramento. A linguagem foi concebida no final dos anos 1980 por Guido van Rossum, com o objetivo de ser uma linguagem de programação simples, legível e fácil de aprender.

Com o passar do tempo, Python ganhou popularidade e começou a ser adotado em uma ampla gama de aplicações e setores. Uma das principais razões para o sucesso de Python é sua ampla comunidade de desenvolvedores, que contribuem para o crescimento da linguagem através do desenvolvimento de bibliotecas e frameworks poderosos.

O ecossistema de bibliotecas Python é extremamente vasto e diversificado. Bibliotecas como NumPy, Pandas, Matplotlib e TensorFlow permitem a realização de tarefas complexas de análise de dados, visualização, machine learning e inteligência artificial. Essas bibliotecas, aliadas à simplicidade e clareza da sintaxe de Python, tornam a linguagem uma escolha popular para projetos de ciência de dados.

Além disso, Python também é amplamente utilizado no desenvolvimento web, graças a frameworks como Django e Flask, que fornecem uma estrutura sólida e eficiente para a construção de aplicativos web escaláveis e robustos. Esses frameworks facilitam a criação de APIs, sistemas de gerenciamento de conteúdo, aplicativos de comércio eletrônico e muito mais.

Outro campo em que Python se destaca é a automação de tarefas. Sua sintaxe concisa e sua ampla variedade de bibliotecas tornam Python uma escolha popular para a automação de processos rotineiros, como extração e manipulação de dados, integração de sistemas e tarefas de gerenciamento de arquivos.

Além disso, Python também é utilizado em áreas como bioinformática, desenvolvimento de jogos, análise financeira, segurança da informação e muito mais. Sua flexibilidade e adaptabilidade permitem que seja aplicado em praticamente qualquer domínio.

## 4. Projeto: Calculadora de Área e Perímetro:

Descrição do Projeto: O projeto consiste em desenvolver uma calculadora de área e perímetro para diferentes formas geométricas, como retângulo, quadrado e círculo. Utilizaremos conceitos de orientação a objetos para criar classes representando cada forma geométrica e métodos para realizar os cálculos necessários.

Passo a Passo do Projeto:

- Criar uma classe base chamada `FormaGeometrica` para representar uma forma geométrica genérica, com atributos de base e altura, e métodos `calcular_area()` e `calcular_perimetro()`.
- Criar classes filhas para cada forma geométrica específica, como `Retangulo`, `Quadrado` e `Circulo`, que herdam da classe base `FormaGeometrica`. Cada classe filha deve implementar os métodos `calcular_area()` e `calcular_perimetro()` conforme as fórmulas matemáticas correspondentes a cada forma.
- No programa principal, criar instâncias das classes `Retangulo`, `Quadrado` e `Circulo`, informando os valores necessários para base, altura e raio, respectivamente. Em seguida, chamar os métodos `calcular_area()` e `calcular_perimetro()` para obter os resultados desejados.

### 4.1. Código do Programa Modelo

```
class FormaGeometrica:
    def __init__(self, base, altura):
        self.base = base
        self.altura = altura

    def calcular_area(self):
        pass

    def calcular_perimetro(self):
        pass

class Retangulo(FormaGeometrica):
    def calcular_area(self):
        return self.base * self.altura

    def calcular_perimetro(self):
        return 2 * (self.base + self.altura)
```

```
class Quadrado(FormaGeometrica):
    def calcular_area(self):
        return self.base ** 2

    def calcular_perimetro(self):
        return 4 * self.base

class Circulo(FormaGeometrica):
    def calcular_area(self):
        return 3.14 * (self.base/2) ** 2

    def calcular_perimetro(self):
        return 2 * 3.14 * (self.base/2)

# Programa principal
retangulo = Retangulo(5, 3)
area_retangulo = retangulo.calcular_area()
perimetro_retangulo = retangulo.calcular_perimetro()

quadrado = Quadrado(4, 0) # Base e altura são iguais para um quadrado
area_quadrado = quadrado.calcular_area()
perimetro_quadrado = quadrado.calcular_perimetro()

circulo = Circulo(6, 0) # Base representa o diâmetro do círculo
area_circulo = circulo.calcular_area()
perimetro_circulo = circulo.calcular_perimetro()

# Exibindo os resultados
print("Retângulo:")
print("Área:", area_retangulo)
print("Perímetro:", perimetro_retangulo)

print("\nQuadrado:")
print("Área:", area_quadrado)
print("Perímetro:", perimetro_quadrado)

print("\nCírculo:")
print("Área:", area_circulo)
print("Perímetro:", perimetro_circulo)
```

## 5. Conclusão:

O projeto da calculadora de área e perímetro é um exemplo prático do poder da linguagem Python e da aplicação dos conceitos de orientação a objetos. Através desse projeto, pudemos explorar e aplicar os princípios fundamentais da programação orientada a objetos, como herança, polimorfismo e encapsulamento.

Ao utilizar Python, uma linguagem conhecida por sua simplicidade e legibilidade, conseguimos criar uma calculadora versátil e de fácil manutenção. Através da criação de classes base e classes filhas específicas para cada forma geométrica, conseguimos reutilizar código e evitar repetições, tornando nosso projeto mais eficiente e escalável.

Além disso, a orientação a objetos nos permitiu separar as responsabilidades de cada classe, tornando o código mais organizado e modular. Isso facilita a manutenção e a adição de novas funcionalidades no futuro.

O projeto da calculadora de área e perímetro é apenas um exemplo das infinitas possibilidades que Python e a orientação a objetos oferecem. Através do domínio desses conceitos, podemos criar soluções robustas, flexíveis e de alto desempenho em diversas áreas, desde desenvolvimento web até ciência de dados e inteligência artificial.

Portanto, esse projeto nos proporcionou uma experiência prática valiosa, permitindo-nos aplicar o conhecimento adquirido sobre Python e orientação a objetos de forma concreta. Com isso, estamos preparados para enfrentar novos desafios e explorar ainda mais o potencial dessa poderosa combinação.



## 6. Bibliografia:

- Python.org. (2021). The Python Programming Language. Recuperado de <https://www.python.org/>
- Van Rossum, G. (2020). The History of Python: A Brief Introduction. Recuperado de <https://www.python.org/doc/essays/foreword/>
- Al Sweigart. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners." No Starch Press, 2015.

*Observação: O código apresentado neste trabalho é um exemplo simplificado e pode ser aprimorado e adaptado para atender a diferentes necessidades. Recomenda-se aprofundar os estudos em Python para explorar seu potencial completo.*