

figures

Дата создания: 06.11.2024, 10:05

Задание: Реализация набора фигур с использованием ООП

Цель: Разработать систему классов для работы с геометрическими фигурами, используя основные принципы объектно-ориентированного программирования: наследование, абстракция, инкапсуляция и полиморфизм.

Требования

1. Создать интерфейс (базовый класс) для всех фигур:

- Интерфейс должен содержать методы для расчета площади (`area()`) и периметра (`perimeter()`), которые будут реализованы в классах-потомках.
- • Интерфейс может быть абстрактным классом (с использованием `abc` и `ABC`).

2. Реализовать классы для не менее чем пяти фигур:

- Квадрат
- Прямоугольник
- Треугольник
- Окружность
- Дополнительная фигура на выбор (например, Ромб, Трапеция и т.д.)

3. Требования к каждому классу фигуры:

- Каждая фигура должна иметь необходимые атрибуты. Например:
- Квадрат: длина стороны.
- Прямоугольник: длина и ширина.
- Треугольник: длина трех сторон или основание и высота.
- Окружность: радиус.
- Каждая фигура должна реализовывать методы для расчета площади и периметра в соответствии со своей геометрией.
- Убедитесь, что атрибуты инкапсулированы (например, имеют защитные или приватные модификаторы).

4. Использование полиморфизма:

- Создайте список, содержащий экземпляры всех фигур, и напишите код, который проходит по списку и вызывает для каждой фигуры методы `area()` и `perimeter()`.
- Демонстрация полиморфизма: программа должна одинаково обращаться со всеми фигурами, несмотря на то, что у каждой из них своя реализация методов.

5. Вопрос к обсуждению:

- Есть ли единый начальный класс, подходящий для всех фигур, или лучше создать отдельные классы для каждой фигуры?
 - Рассмотрите, какие атрибуты и методы могли бы быть общими для всех фигур, а какие — уникальными для каждой конкретной фигуры. Как можно организовать иерархию наследования для гибкости и расширяемости?
-

Теги

Python