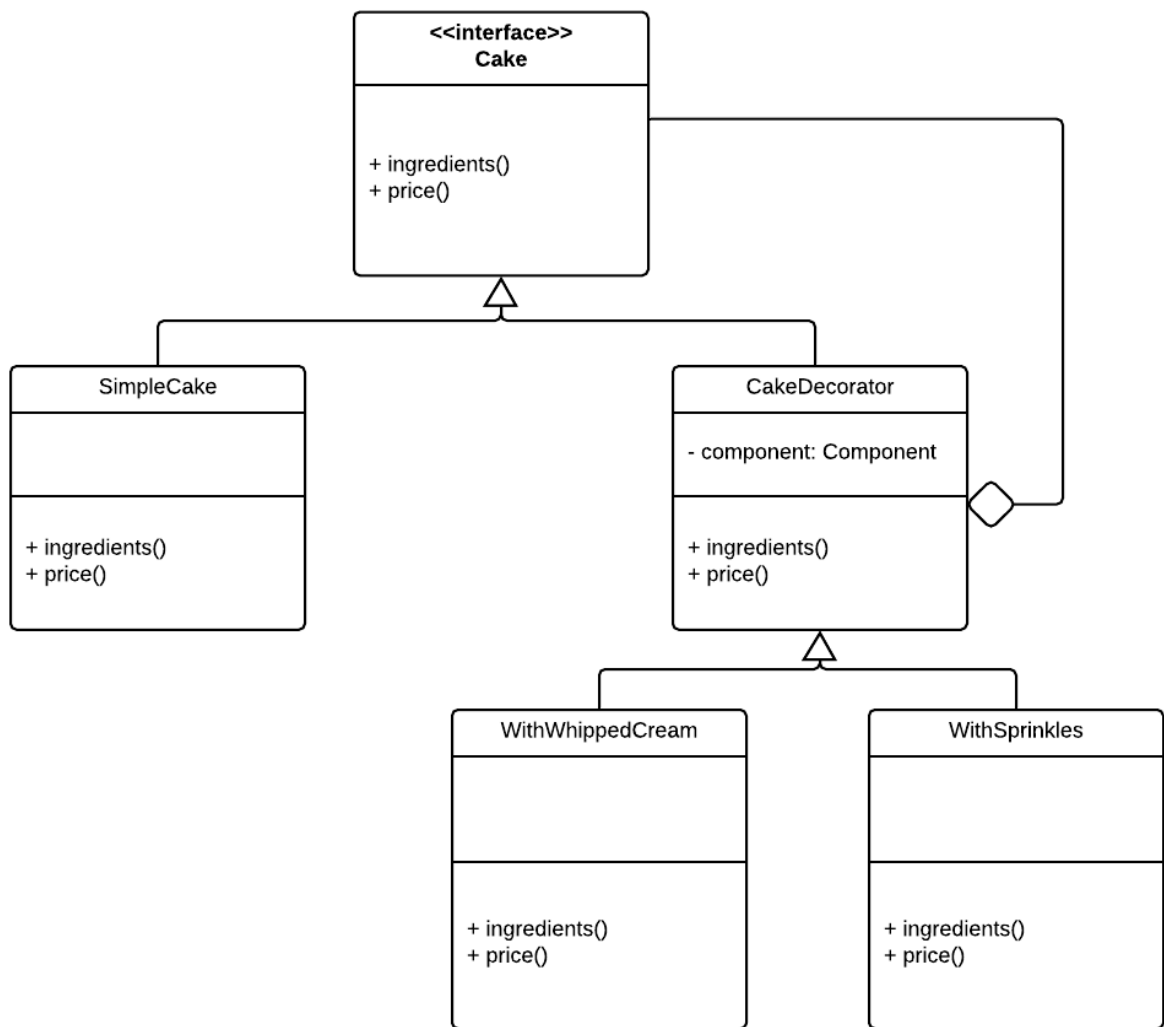


**Паттерн Декоратора** — добавляет поведение экземпляру класса, не влияя на поведение других экземпляров того же класса. Может это делать динамически.

Этот паттерн часто используется как альтернатива подклассам. Декорирование может обеспечить новое поведение для объектов во время выполнения, в то время как подклассы добавляют поведение во время компиляции и влияют на все экземпляры исходного класса. Эта различие между декорированием и созданием подклассов становится еще более важной, когда существует множество независимых способов расширения функциональности.

Еще одной причиной, по которой часто используется Декоратора, является соблюдение принципа единой ответственности, поскольку он позволяет инкапсулировать функциональность в различные классы декоратора.



```

interface Cake
{
    public function ingredients(): string;
    public function price(): double;
}

abstract class CakeDecorator implements Cake
{
    /**
     * @var Cake
     */
    protected $cake;
    public function __construct(Cake $cake)
    {
        $this->cake = $cake;
    }
}

class SimpleCake implements Cake
{
    public function ingredients(): string
    {
        return 'Простой торт';
    }
    public function price(): double
    {
        return 12.5;
    }
}

```

```

class WithWhippedCream extends CakeDecorator
{
    public function ingredients(): string
    {
        return $this->cake->ingredients() + ' со взбитыми сливками';
    }
    public function price(): double
    {
        return $this->cake->price() + 2.5;
    }
}

```

```

class WithSprinkles extends CakeDecorator
{
    public function ingredients(): string
    {
        return $this->cake->ingredients() + ' с обсыпкой';
    }
    public function price(): double
    {
        return $this->cake->price() + 1.25;
    }
}

```

```

$cake = new SimpleCake();
$cake = new WithWhippedCream($cake);
$cake = new WithSprinkles($cake);
echo $cake->ingredients(); // Простой торт со взбитыми сливками с обсыпкой
echo $cake->price(); // 16.25
// или можно цепочкой
$cake = new WithSprinkles(new WithWhippedCream(new SimpleCake()));
echo $cake->ingredients(); // Простой торт со взбитыми сливками с обсыпкой
echo $cake->price(); // 16.25

```