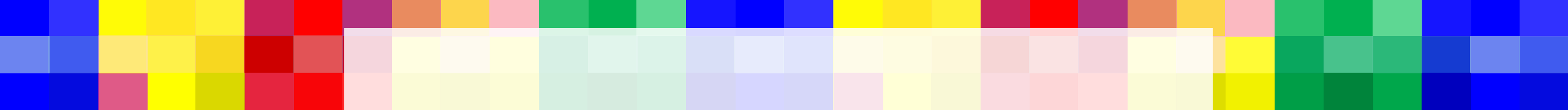


Паттерны

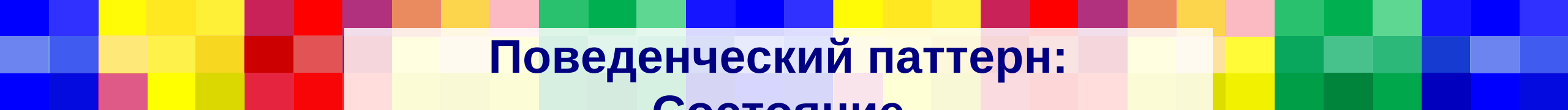


- 
- ❑ порождающие паттерны — используются для создания объектов с конкретным поведением;
 - ❑ структурные паттерны — помогают структурировать код для конкретных случаев применения;
 - ❑ поведенческие паттерны — помогают распределить обязанности и инкапсулировать поведение.



Поведенческий паттерн: шаблонный метод

- это поведенческий паттерн проектирования, который определяет скелет алгоритма, перекладывая ответственность за их реализацию некоторых из них на подклассы.



Поведенческий паттерн: Состояние

- о поведенческий паттерн проектирования, который позволяет объектам менять поведение в зависимости от своего состояния.



Поведенческий паттерн: Стратегия

- определяет общее поведение для конкретных алгоритмов, реализующих разные стратегии



Порождающий паттерн: Синглтон

- Гарантирует создание единого экземпляра класса



Порождающий паттерн: Абстрактная фабрика

- порождающий шаблон проектирования, предоставляет интерфейс для создания семейств взаимосвязанных или взаимозависимых объектов, не специфицируя их конкретных классов.



Порождающий паттерн: Прототип

- это порождающий паттерн проектирования, который позволяет копировать объекты, не вдаваясь в подробности их реализации.




Структурный паттерн: Декоратор

- Дополнить функционал отдельному объекту, а не на классе в целом.



@staticmethod

- Статический метод в Python — это метод, который принадлежит классу, а не экземпляру класса. Он не может модифицировать состояние класса или его экземпляров. Статические методы объявляются с помощью декоратора `@staticmethod`.
 - Преимущество статических методов в том, что они могут быть вызваны даже если нет объекта класса. Они также могут быть переопределены в подклассах.
- 

- Метод класса в Python — это метод, который привязан к классу, а не к его экземпляру. Он может изменять состояние класса, но не может изменять состояние конкретного экземпляра класса. Методы класса объявляются с помощью декоратора `@classmethod`.
- Методы класса принимают ссылку на класс, который вызывает метод, в качестве первого аргумента (обычно именуемого `cls`). Их основное преимущество в том, что они могут быть наследованы и переопределены в подклассах.