

## Bridge

The Bridge pattern facilitates the combination of objects into a larger structure, illustrating a hierarchy of components. This pattern is versatile, finding application in scenarios such as depicting varied implementations, frameworks, or platforms. It allows clients to handle individual objects and compositions of objects uniformly, promoting a more generic and simplified client code.

The key participants in the Bridge pattern include:

**Abstraction:** Declares an interface for the Abstraction part of the bridge. Defines default behavior and maintains a reference to the Implementor.

**Refined Abstraction:** Extends the abstraction by building on the basic operations provided by the Abstraction.

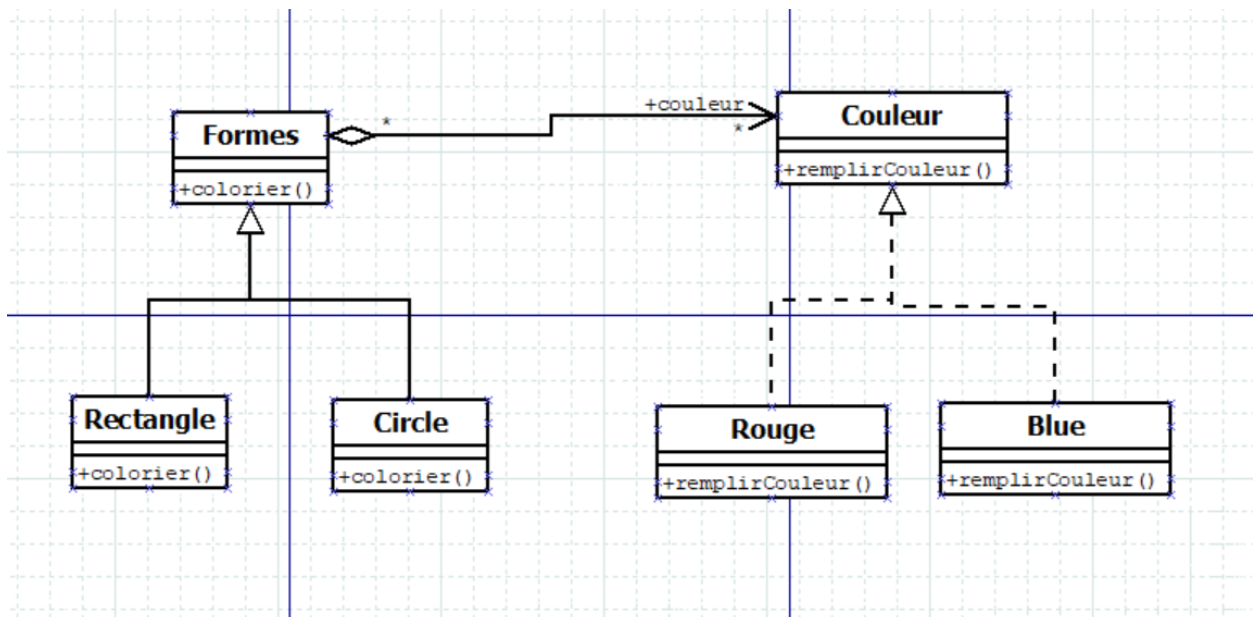
**Implementor:** Declares an interface for the Implementor part of the bridge, defining the primitive operations.

**Concrete Implementor:** Provides specific implementations for the primitive operations defined by the Implementor.

### 1. Initial

Here I am implementing the general structure and the first example of the found in the pdf we were taught with in class.

### Model



## 2. Modified

Here, I took the initial code from the pdf and modified it to add a **new colour** and a **new shape**.

### Model

