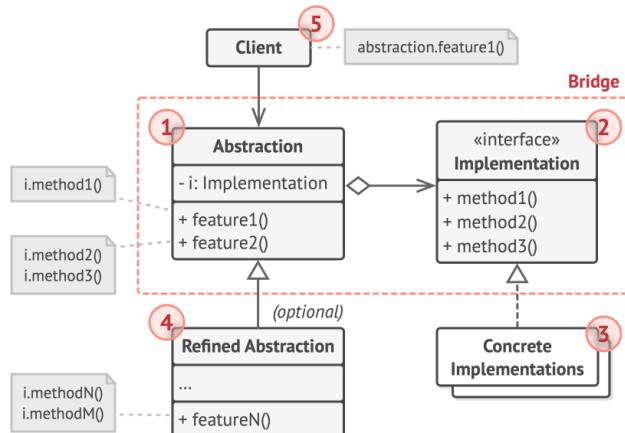


# Bridge

E' un pattern strutturale che consente di suddividere una classe di grandi dimensioni o un insieme di classi strettamente correlate in due gerarchie distinte, astrazione e implementazione, che possono essere sviluppate indipendentemente l'una dall'altra.

- Astrazione = è un livello di controllo di alto livello per un'entità (interfaccia). Tale livello non svolge alcun lavoro reale. Dovrebbe delegare il lavoro al livello di implementazione (piattaforma).

## Struttura



- L'astrazione fornisce una logica di controllo di alto livello. Si affida all'oggetto di implementazione per svolgere il lavoro effettivo di basso livello.
- L'implementazione dichiara l'interfaccia comune a tutte le implementazioni concrete. Un'astrazione può comunicare con un oggetto di implementazione solo tramite i metodi dichiarati qui.
- Le implementazioni concrete contengono codice specifico della piattaforma;
- Le astrazioni raffinate forniscono varianti della logica di controllo;
- Il cliente è interessato solo a lavorare con l'astrazione.

## **Applicabilità**

- Usare Bridge quando si desidera dividere e organizzare una classe monolitica che presenta diverse varianti di alcune funzionalità;
- Usare il modello quando è necessario estendere una classe in più dimensioni ortogonali;
- Usare Bridge se è necessario cambiare implementazione in fase di esecuzione;

## **Svantaggi**

- Complessità del codice applicando il modello a una classe altamente coesa;

## **Vantaggi**

- E' possibile creare classi e app indipendenti dalla piattaforma;
- Il codice client funziona con astrazioni di alto livello;
- Principio open/closed;
- Principio di singola responsabilità.

## **Relazioni con altri modelli**

- Bridge viene usato in anticipo, in modo da sviluppare parti di un'applicazione indipendentemente l'una dall'altra. Adapter, invece, viene usato con un'app esistente per far funzionare insieme classi incompatibili;
- Bridge, State, Strategy e in parte Adapter hanno strutture molto simili. Ma risolvono problemi diversi;
- E' possibile usare Abstract Factory insieme a Bridge. Quando alcune astrazioni definite da Bridge possono funzionare solo con implementazioni specifiche;

- E' possibile combinare Builder e Bridge. la classe director svolge il ruolo di astrazione, mentre i diversi builder fungono da implementazioni.