Card Class   -   The Bridge Pattern

**Abstraction**

**Implementor**

Card

+ logic : CardLogic

+ placeCardOnTile(tile : Tile)
+ getDescription() : String
+ useCardOnPlayer(player : Player)

0..*

<<interface>>
**CardLogic**

+placeCardOnTile(card : Card, tile : Tile)
+useCardOnPlayer(card : Card, player : Player)

**Player**

+ hand : ArrayList<Card>

**Board**

+ deck : Stack<Card>

ActionCard

PathCard

PersonalCard

**Refined Abstractions**

PathCardLogic

ActionCardLogic

ChanceCardLogic

HealCardLogic

... etc.

**Concrete Implementors**

**DeadEndCard**

**StraightCard**

**HostageCard**
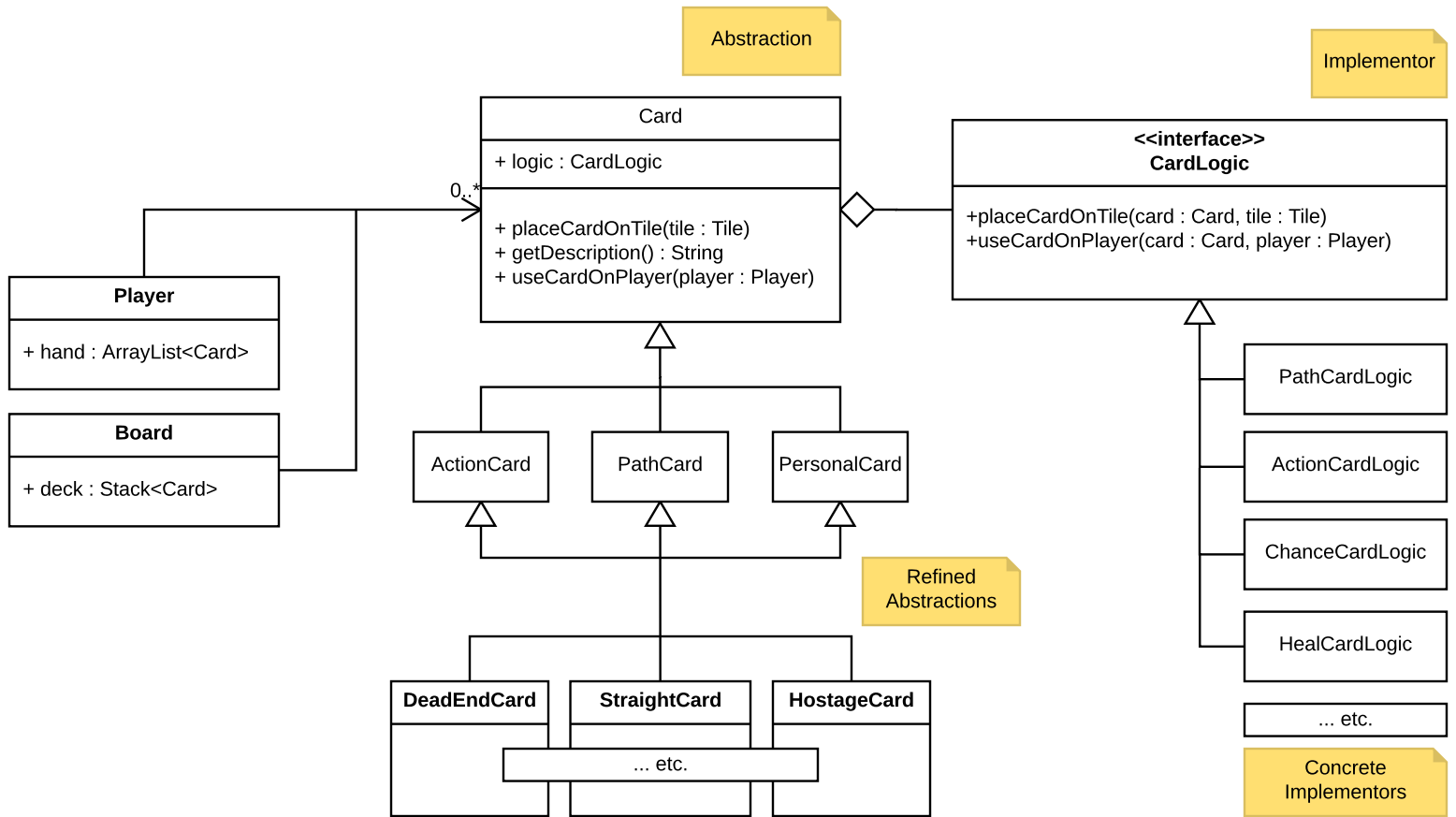
... etc.

Bridge Pattern  -  Structural - Decouple an abstraction from its implementation so that the two can vary independently.
Calling a function in Card will call the appropriate function in its CardLogic.

```
                                    ┌─────────────────────────────┐
                                    │        GameContext          │
                                    ├─────────────────────────────┤
                                    │                             │
                                    │ +commHistory:CommandHistory │
                                    │                             │
                                    └─────────────────────────────┘
                                                  │
                                                  ▼
┌─────────────────────────────┐     ┌─────────────────────────────┐
│        <<interface>>        │     │       CommandHistory        │
│          Command            │◄────├─────────────────────────────┤
├─────────────────────────────┤     │ +undoStack:Stack<Command>   │
│ +executeCommand()           │     │ +undoLast()                 │
│ +undo()                     │     └─────────────────────────────┘
└─────────────────────────────┘
              △
    ┌─────────┼─────────┐
    │         │         ▼
┌────────┐ ┌──────────┐ ┌──────────────┐
│Place   │ │Discard   │ │DonateCommand │
│Command │ │Command   │ │              │
└────────┘ └──────────┘ └──────────────┘
```