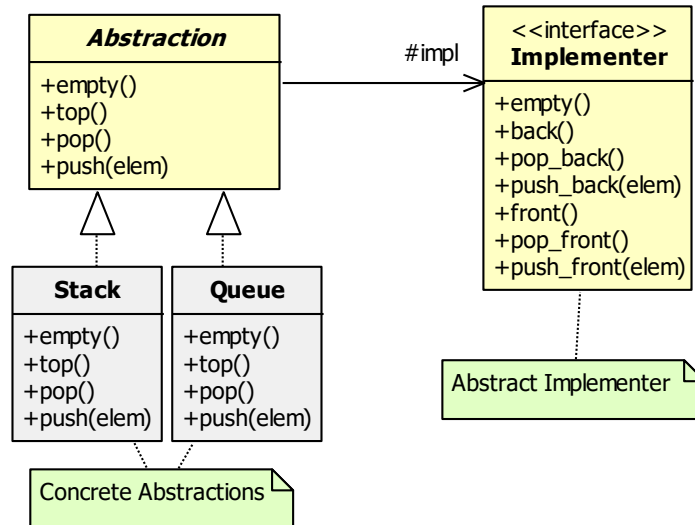


Problem

Դիտարկենք **Bridge (GoF)** շաբլոնի օգտագործման հետևյալ օրինակը.



Ներկայացնենք այս կառուցվածքի հիմքում ընկած գաղափարը ընդհանրացված ծրագրավորման հենքի վրա:

- Սահմանենք *Implementer* կոնցեպցիա հետևյալ թույլատրելի արտահայտություններով.

```
bool empty()const;

T& back()const;           //precondition: !empty()
void pop_back();          //precondition: !empty()
void push_back(const T& elem);

T& front()const;          //precondition: !empty()
void pop_front();         //precondition: !empty()
void push_front(const T& elem);
```

- Ներկայացնենք *Concrete Abstraction*-ները որպես պարամետրիզացված դասեր.

```
template< class T, class Implementer> class Stack {...};
template< class T, class Implementer> class Queue {...};
```

Խնդիր:

- ✓ Սահմանել *Implementer* կոնցեպցիայի մոդել հանդիսացող **List<T>** դաս՝ որն իրականացնում է կոնցեպցիայի գործողությունները երկկապ ցուցակի հիման վրա:
- ✓ Իրականացնել **Stack<T, Implementer>** և **Queue<T, Implementer>** դասերը՝ հենվելով *Implementer* կոնցեպցիայի թույլատրելի արտահայտությունների վրա:
- ✓ Օգտագործել **bridge_test.cpp** ֆայլը կառուցված կոդը թեստավորելու համար: