# DESIGN FULL EVENT

We will do several Hashmap to store the different informations we need.

- Map<Integer, AcceptListener> port
- Map<Integer, ArrayList<BrokerWithListener>> brokerwithListener
- Map<Integer, ArrayLIst <Channel>> channels

## Disconnected

If there is no connection before ,return false.
Else we will be in a runnable and we will post an event to the executor.
In that event, we will close all the channels .

## Connect

If the remote broker doesn't exist, return false.
Else we will be in a runnable and we will post an event to the executor.
In that event,if there is already an accept, we will create a channel and the remote channel and we will call the listener of the accept and the listener of the connect.
Else we will add to the brokerWithListener the broker and the listener if this is already created else we will create it and we will add to the brokerWithListener the broker and the listener.

## Accept

If there is already a accept, return false.
Else we will be in a runnable and we will post an event to the executor.
In that event,if the brokerWithListener is not empty, we will create a channel and the remote channel and we will call the listener of the accept and the listener of the connect.
Else we post an event to the executor to wait for a connect.

#################

## read

If the channel is closed or the remote channel is closed and the buffer is empty, we will throw a ClosedException.
Else we will be in a runnable and we will post an event to the executor.
In that event,if the buffer is empty, we will add the message to a queue and we will post an event to the executor to read the message.
Else we will read the message and we will call the listener of the writer if the the queue of the writer is not empty.

## write

If the channel is closed,we will throw a ClosedException.
If the remote channel is closed, we will call the listener of the writer and return true.
Else we will be in a runnable and we will post an event to the executor.
In that event,if the buffer is full, we will add the message to a queue and we will post an event to the executor to write the message.
Else we will write the message and we will call the listener of the reader if the the queue of the reader is not empty.