(def context1 {:state-type "Established"})

(def context2 {:state-type "Listen"})

(def context3 {:state-type "Closed"})

(defmulti **open-tcp-connection** (fn [context] (:state-type context)))

(defmethod open-tcp-connection "Established" [context]

(println "Opening tcp connection for state" (:state-type context) "...")

)

(defmethod open-tcp-connection "Listen" [context]

(println "Opening tcp connection for state" (:state-type context) "...")

)

(defmethod open-tcp-connection "Closed" [context]

(println "Opening tcp connection for state" (:state-type context) "...")

)

(defmulti **close-tcp-connection** (fn [context] (:state-type context)))

(defmethod close-tcp-connection "Established" [context]

(println "Closing tcp connection for state" (:state-type context) "...")

)

(defmethod close-tcp-connection "Listen" [context]

(println "Closing tcp connection for state" (:state-type context) "...")

)

(defmethod close-tcp-connection "Closed" [context]

(println "Closing tcp connection for state" (:state-type context) "...")

)

(defmulti **ack-tcp-connection** (fn [context] (:state-type context)))

(defmethod ack-tcp-connection "Established" [context]

(println "Acknowledging tcp connection for state" (:state-type context) "...")

)

(defmethod ack-tcp-connection "Listen" [context]

(println "Acknowledging tcp connection for state" (:state-type context) "...")

)

(defmethod ack-tcp-connection "Closed" [context]

(println "Acknowledging tcp connection for state" (:state-type context) "...")

)

(**open-tcp-connection** context1)

(**close-tcp-connection** context2)

(**ack-tcp-connection** context3)