

# State

---



**Karoly Nyisztor**

DEVELOPER

@knyisztor [www.leakka.com](http://www.leakka.com)

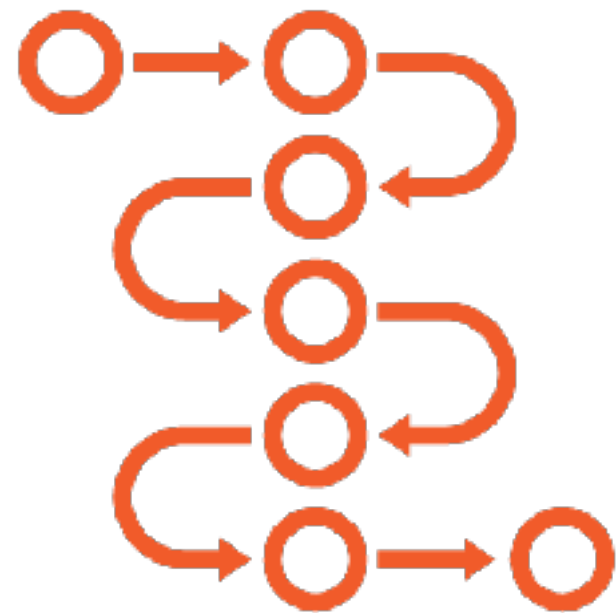
# Overview

## **Motivation**

## **ATM demo case study**

- Refactor using the State design pattern

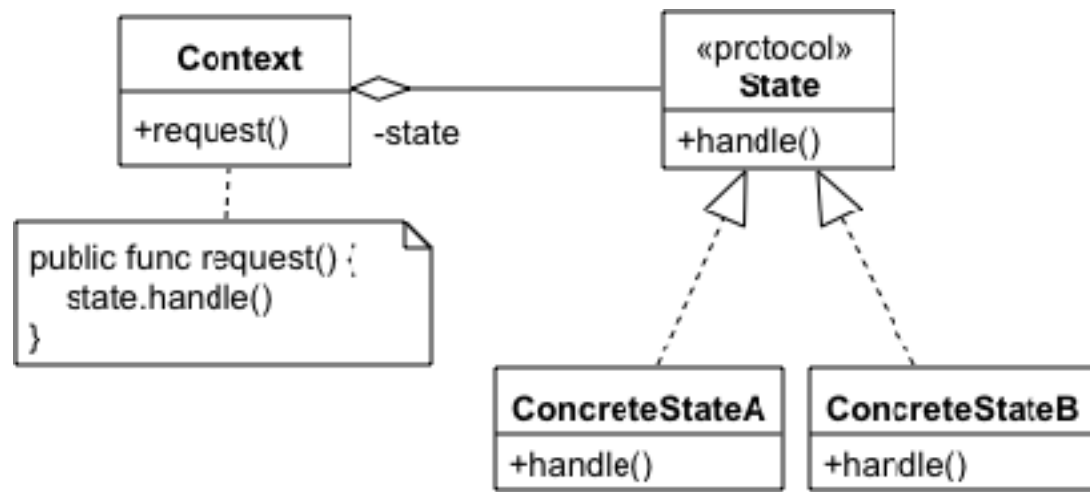
# Motivation



## Object-oriented state machine

- Remove monolithic conditional logic
- Encapsulate state behavior into dedicated types
- Increase flexibility
- Reduce maintenance efforts

# State Design



## Context

- Exposes client-side interface.
- Delegates requests to its state

## State

- Defines a common interface for all states

## Concrete state

- Implements behavior associated with a state

# State

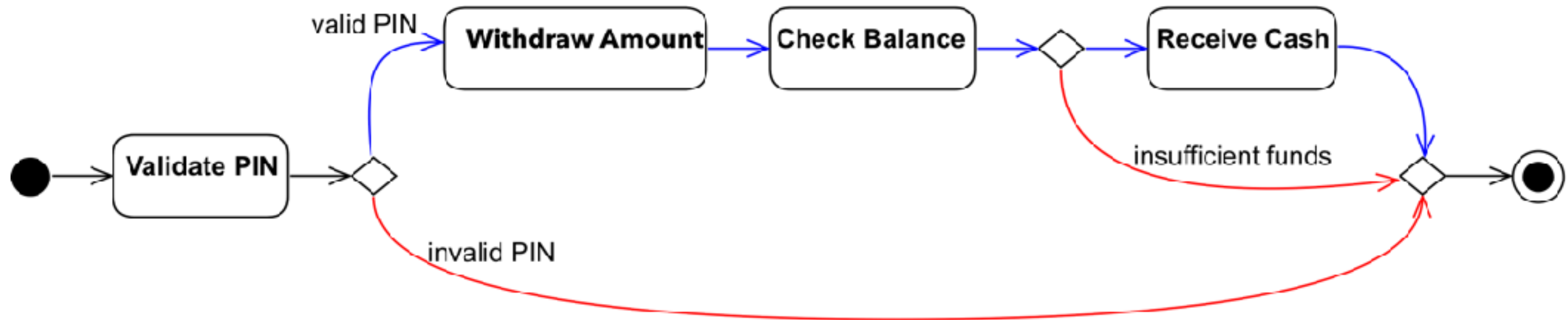
Allows an object to behave differently when its internal state changes.

# Demo

## **ATM demo refactor**

- Revamp using the State pattern
- Get rid of complex conditional logic

# ATM Activity Diagram



# Summary

## **The State design pattern:**

- Allows objects to behave differently as their internal state changes
- Avoids monolithic conditional logic
- Encapsulates state behavior into state types

## **Pitfalls**

- Exposing State types to clients
- Tight coupling between state types