Group 6:

Lê Duy Bách 1551003

Đào Thanh Danh 1551004

Nguyễn Khắc Tuấn 1551043

CS 202: PROGRAMMING SYSTEM

SEMINAR DOCUMENT

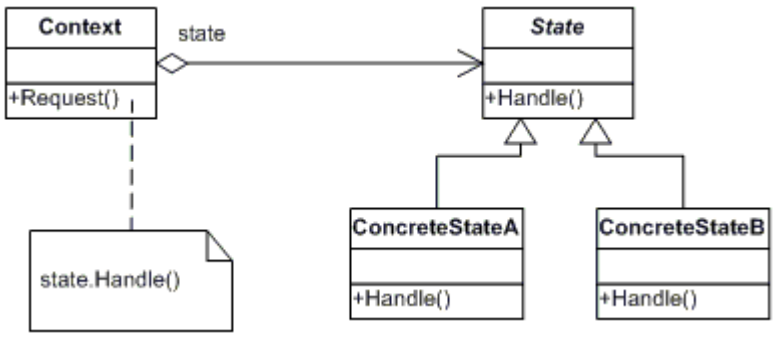
STATE DESIGN PATTERN

I. Introduction:

- State is one of the behavior patterns. In State, class behavior varies upon its state. In State pattern, we create objects which represent various states and a context object whose behavior is adjusted depending on its state object changes.

II. Components:

- Context class
  - This class is the machine itself.
  - This class will invoke request(void)
- State class
  - Abstract class of the state
  - State* -> Handle(void)
  - Contain virtual function which will be overriden by the concrete state object
- Concrete State A
  - Specific state of the context object
  - Override state::Handle(void)
- Concrete State B
  - Similar to Concrete State A



Class diagram of State pattern

III. Example:

- Situation: An ATM Machine with and without card inserted will respond differently upon button pressed
- Approach:
  - Context class: ATM Machine
  - State class: Card status
  - Concrete class A: CardInside
  - Concrete class B: CardOutside
- Action:
  - Context class method: Machine.buttonPress( ) { state->respond; }
  - State class: virtual void respond(void) = 0;

- Concrete State A (CardInside): void respond( ) { text += btn; }
- Concrete State B (CardOutside): void respond( ) { display("No card"); }
- Number of states can increase depend on the actual problem!

IV. Reference:

https://sourcemaking.com/design_patterns/state/cpp/1
https://sourcemaking.com/design_patterns/state
http://www.journaldev.com/1751/state-design-pattern-java