Java-20\_Lab9

32191826 박주희

<State Pattern>

State Pattern이란 객체가 특정 상태에 따라 행위를 달리하는 상황에서, 자신이 직접 상태를 체크하여 상태에 따라 행위를 호출하지 않고 상태를 객체화 하여 상태가 행동을 할 수 있도록 위임하는 패턴이다.

🡪 객체의 특정 상태를 클래스로 선언하고, 클래스에서는 해당 상태에서 할 수 있는 행위들을 메서드로 정의한다. 그리고 이러한 각 상태 클래스들을 인터페이스로 캡슐화 하여, 클라이언트에서 인터페이스를 호출하는 방식이다.

<수업시간에 배운 GumballWithState>

class **HasQuarterState, NoQuarterState, SoldOutState, SoldState** implements **State**

insertQuarter(), ejectQuarter(), turnCrank(), dispense(), refill() **🡪 State**를 implement하는 모든 클래스들이 구현해야 하는 메소드

class **GumballMachineWithState** 🡪 생성자에서 모든 State class 객체를 생성하고 이를 get

상태 인터페이스의 메서드를 호출하면, 각 상태 클래스에서 정의된 행위가 수행되는 방식

<전체적인 구조>

class **StartState, IntegerState, DecimalState** implements **State**

plus(), minus(), dot(), digit(), other() 🡪 각각 +, -, ., 숫자, 나머지 상태

class **DoubleParser** 🡪 char형을 parse하여 double로 반환

ex) “123”

‘1’ 🡪 1 ‘2’🡪1\*10+2=12 ‘3’🡪12\*10+3=123

🡪 제대로 된 수를 입력하면 그대로 출력되고 잘못된 수식을 입력할 경우 error 메시지가 뜬다.

<주요 코드와 구현과정>

비교적 쉬었던 과제이다. 교수님께서 Finite State Machine을 이해하기 쉽게 그림으로 제공해주셔서 수월하게 할 수 있었다.



init()는 초기화를 하는 함수이고 while문을 이용하여 입력받은 str을 하나씩 자르고 이에 따른 state.상태를 호출한다. while문이 끝날 때까지를 이를 반복하고 계산값을 반환한다. check 변수는 한번이라도 error가 발생했는지 검사하는 boolean이다. 오류가 한번도 없었을 때는 계산값이 반환되고 오류가 발생되었으면 0.0이 반환된다.



DoubleParser에서 State state=new StartState(this); 를 하였기 때문에 처음 입력될 때 state는 StartState이다. 처음에 +23 또는 -23처럼 +와 -가 나올 수 있는데 이때 sign을 각각 1, -1이라고 한다. 이로인해 마지막에 toDouble()의 결과값 또한 부호가 제대로 나올 수 있는 것이다.

plus(), minus()일 경우에는 입력 숫자 값이 integer라고 판단되므로 dp.setState(dp.getIntegerState())를 추가해주고 dot()일 경우에는 소수라고 판단되므로 dp.setState(dp.getDemicalState())를 추가해준다.

처음에 +, -, ., 숫자 이외의 것이 나오면 숫자가 아니므로 error 메시지가 뜨도록 한다.

이처럼 다른 상태 클래스에서도 숫자, 소수, 오류를 판단하여 코드를 작성해준다.



배열에 먼저 저장해놓고 for문을 이용하여 doubleparser.toDouble()의 매개변수로 모든 원소들을 넣어주어 결과를 출력하였다.

<My code >



Jframe을 이용하여 입력칸을 만들고 button을 누르면 doubleparser 값이 나오도록 하였다.

<결과화면>

