Writing the code

It's time to implement the Gumball Machine. We know we're going to have an instance variable that holds the current state. From there, we just need to handle all the actions, behaviors and state transitions that can happen. For actions, we need to implement inserting a quarter, removing a quarter, turning the crank and dispensing a gumball; we also need the empty gumball condition to implement as well.

```java
public class GumballMachine {  
  final static int SOLD_OUT = 0;
  final static int NO_QUARTER = 1;
  final static int HAS_QUARTER = 2;
  final static int SOLD = 3;

  int state = SOLD_OUT;
  int count = 0;

  public GumballMachine(int count) {
    this.count = count;
    if (count > 0) {
      state = NO_QUARTER;
    }

    // Now we start implementing the actions in methods...

    public void insertQuarter() {  
      if (state == HAS_QUARTER) {
        System.out.println("You can't insert another quarter");
      } else if (state == NO_QUARTER) {
        System.out.println("You inserted a quarter");
        state = HAS_QUARTER;
      } else if (state == SOLD) {
        System.out.println("You can't insert a quarter, the machine is sold out");
      } else if (state == SOLD_OUT) {
        System.out.println("Please wait, we're already giving you a gumball");
      }
      if (this.count > 0) {
        this.count--;
      }
    }

    public void turnCrank() {  
      if (state == SOLD) {
        System.out.println("Turning twice doesn't get you another gumball!");
      } else if (state == NO_QUARTER) {
        System.out.println("You turned but there's no quarter");
      } else if (state == SOLD_OUT) {
        System.out.println("You turned, but there are no gumballs");
      } else if (state == HAS_QUARTER) {
        System.out.println("You turned...");
        state = SOLD;
        dispense();
      }
    }

    public void dispense() {  
      if (state == HAS_QUARTER) {
        System.out.println("A gumball comes rolling out the slot");
        count = count - 1;
        if (count == 0) {  
          System.out.println("Ooops, out of gumballs!");
        } else {
          state = SOLD;
        }
      } else if (state == NO_QUARTER) {
        System.out.println("You need to pay first");
      } else if (state == SOLD) {
        System.out.println("No gumball dispensed");
      } else if (state == SOLD_OUT) {
        System.out.println("No gumball dispensed");
      }
      // other methods here like toString() and refill()
      }
    }
  }
  
  // the state pattern

  public void ejectQuarter() {  
    if (state == HAS_QUARTER) {
      System.out.println("Quarter returned");
      state = NO_QUARTER;
    } else if (state == NO_QUARTER) {
      System.out.println("You haven't inserted a quarter");
    } else if (state == SOLD) {
      System.out.println("Sorry, you already turned the crank");
    } else if (state == SOLD_OUT) {
      System.out.println("You can't eject, you haven't inserted a quarter yet");
    }
    }
  
  public void turnCrank() {  
    if (state == SOLD) {
      System.out.println("Turning twice doesn't get you another gumball!");
    } else if (state == NO_QUARTER) {
      System.out.println("You turned but there's no quarter");
    } else if (state == SOLD_OUT) {
      System.out.println("You turned, but there are no gumballs");
    } else if (state == HAS_QUARTER) {
      System.out.println("You turned...");
      state = SOLD;
      dispense();
    }
    }
  
  // called to dispense a gumball
  public void dispense() {  
    if (state == HAS_QUARTER) {
      System.out.println("A gumball comes rolling out the slot");
      count = count - 1;
      if (count == 0) {  
        System.out.println("Ooops, out of gumballs!");
      } else {
        state = SOLD;
      }
    } else if (state == NO_QUARTER) {
      System.out.println("You need to pay first");
    } else if (state == SOLD) {
      System.out.println("No gumball dispensed");
    } else if (state == SOLD_OUT) {
      System.out.println("No gumball dispensed");
    }
    // other methods here like toString() and refill()
  }
  
  // the state pattern

  public void ejectQuarter() {  
    if (state == HAS_QUARTER) {
      System.out.println("Quarter returned");
      state = NO_QUARTER;
    } else if (state == NO_QUARTER) {
      System.out.println("You haven't inserted a quarter");
    } else if (state == SOLD) {
      System.out.println("Sorry, you already turned the crank");
    } else if (state == SOLD_OUT) {
      System.out.println("You can't eject, you haven't inserted a quarter yet");
    }
    }
  
  public void turnCrank() {  
    if (state == SOLD) {
      System.out.println("Turning twice doesn't get you another gumball!");
    } else if (state == NO_QUARTER) {
      System.out.println("You turned but there's no quarter");
    } else if (state == SOLD_OUT) {
      System.out.println("You turned, but there are no gumballs");
    } else if (state == HAS_QUARTER) {
      System.out.println("You turned...");
      state = SOLD;
      dispense();
    }
    }

  // called to dispense a gumball
  public void dispense() {  
    if (state == HAS_QUARTER) {
      System.out.println("A gumball comes rolling out the slot");
      count = count - 1;
      if (count == 0) {  
        System.out.println("Ooops, out of gumballs!");
      } else {
        state = SOLD;
      }
    } else if (state == NO_QUARTER) {
      System.out.println("You need to pay first");
    } else if (state == SOLD) {
      System.out.println("No gumball dispensed");
    } else if (state == SOLD_OUT) {
      System.out.println("No gumball dispensed");
    }
    // other methods here like toString() and refill()
  }
```

390  Chapter 10

391  you are here