

루비 디자인 패턴 : 템플릿 패턴
김대권

propellerheaven@gmail.com

템플릿 패턴

시나리오

- 레포트 생성기

HTML Report

- 간단한 월례 보고서 생성기
- 제목과 본문 필요
- HTML 포맷만 잘 출력

Report 클래스

```
class Report
  def initialize
    @title = 'Monthly Report'
    @text = ['Things are going', 'really, really well.']
  end

  def output_report
    puts('<html>')
    puts(' <head>')
    puts("   <title>#{@title}</title>")
    puts(' </head>')
    puts(' <body>')
    @text.each do |line|
      puts("   <p>#{line}</p> ")
    end
    puts(' </body>')
    puts('</html>')
  end
end
```

Report 클래스(cont.)

```
report = Report.new  
report.output_report
```

하지만 현실은…

다양한 출력 형식?

- 이렇게도 해야하고
- 저렇게도 해야하고
- 나중엔 그렇게도 해야할지도 모르는

월례 리포트 생성기

- HTML 포맷
- Text 포맷

Report 클래스

- 일단은 고치고 보자

수정된 Report 클래스

```
def output_report(format)
  if format == :plain
    puts("*** #{@title} ***")
  elsif format == :html
    puts('<html>')
    puts(' <head>')
    puts("   <title>#{@title}</title>")
    puts(' </head>')
    puts(' <body>')
  else
    raise "Unknown format: #{format}"
  end

  @text.each do |line|
    if format == :plain
      puts(line)
    else
      puts("   <p>#{line}</p> ")
    end
  end

  if format == :html
    puts(' </body>')
    puts('</html>')
  end
```

회고

- if 문을 사용한 분기
- 모든 처리를 outputreport에서 하고 있음
- 뭔가 이게 아닌 것 같은데… 모르겠다
- ...

다시 월례 리포트 생성기

- HTML로 보내주세요.
- Text로도 보내주세요.
- RTF가 필요해요.
- PostScript 포맷이 필요해요.
- ...



템플릿 패턴

레포트 생성 과정

1. Outputs any header information required by the specific format.
2. Output the title.
3. Output each line of the actual report.
4. Output any trailing stuff required by the format.

outputreport Method

```
def output_report
    output_start
    output_head
    output_body_start
    output_body
    output_body_end
    output_end
end
```

Abstract Method

```
def output_start
    raise 'Called abstract method: output_start'
end

def output_head
    raise 'Called abstract method: output_head'
end

def output_body_start
    raise 'Called abstract method: output_body_start'
end

def output_line(line)
    raise 'Called abstract method: output_line'
end

def output_body_end
    raise 'Called abstract method: output_body_end'
end

def output_end
    raise 'Called abstract method: output_end'
end
```

HTML Report

```
class HTMLReport < Report
  def output_start
    puts('<html>')
  end

  def output_head
    puts(' <head>')
    puts("   <title>#{@title}</title>")
    puts(' </head>')
  end

  def output_body_start
    puts('<body>')
  end

  def output_line(line)
    puts(" <p>#{@line}</p>")
  end

  def output_body_end
    puts('</body>')
  end

  def output_end
    puts('</html>')
```

Text Report

```
class PlainTextReport < Report
  def output_start
  end

  def output_head
    puts("***** #{@title} *****")
    puts
  end

  def output_body_start
  end

  def output_line(line)
    puts line
  end

  def output_body_end
  end

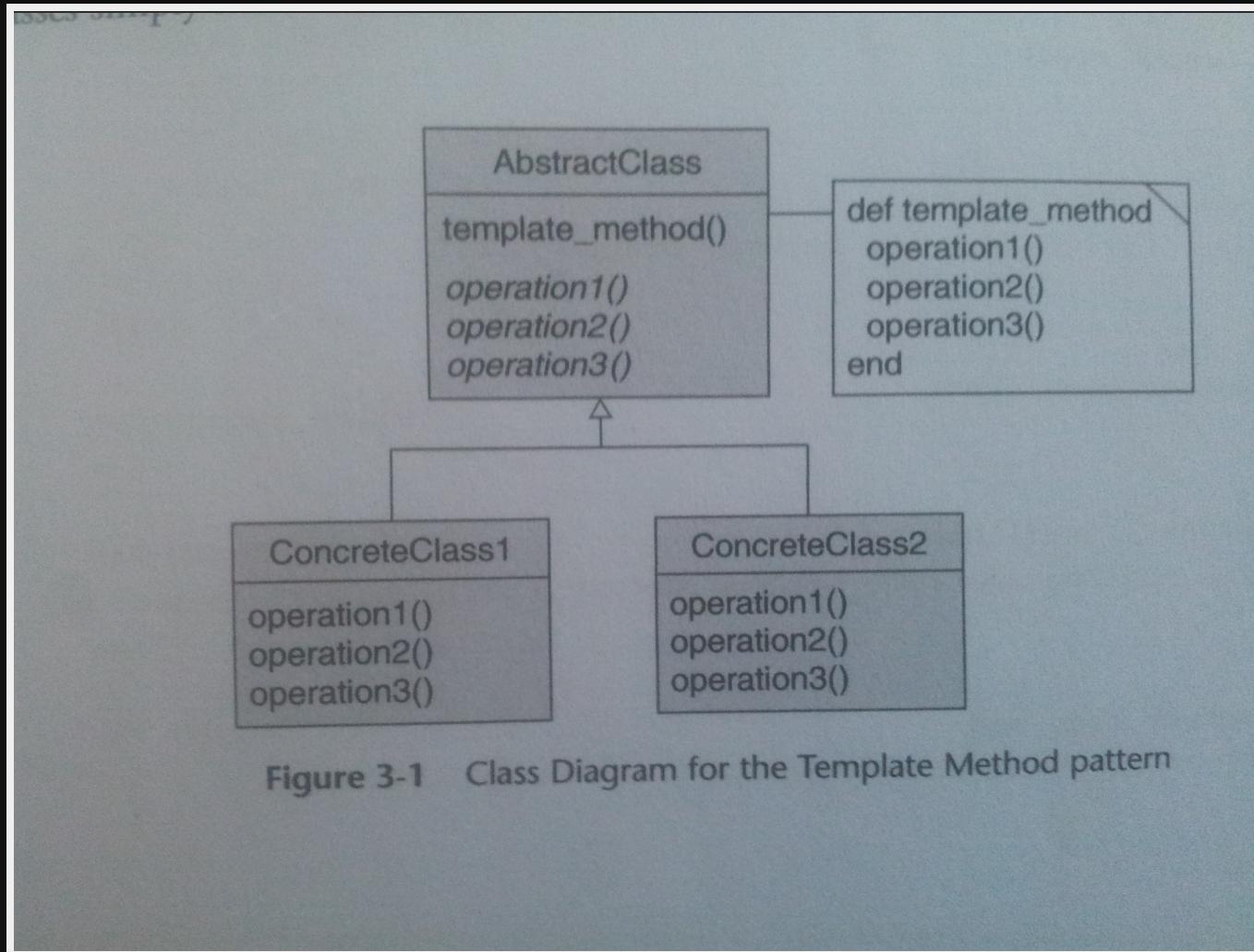
  def output_end
  end
end
```

Report 생성하기

```
report = HTMLReport.new  
report.output_report
```

```
report = PlainTextReport.new  
report.output_report
```

템플릿 패턴 클래스 다이어그램



Hook Methods

- 상속 클래스에서 오버라이드해서 사용할 수도 있음
- 기본 기능을 제공

Report 클래스

```
class Report
...
def output_start
end

def output_body_start
end

...
def output_body_end
end

def output_end
end
end
```

PlainTextReport

- outputbodystart 와 outputbodyend 는 빈 메소드
- HTML 레포트에서 body 태그가 들어가는 메소드

PlainTextReport

```
class PlainTextReport < Report
  def output_start
  end

  def output_head
    puts("***** #{@title} *****")
    puts
  end

  def output_line(line)
    puts(line)
  end

  def output_end
  end
end
```

Duck Typing

- 오리 같이 생겼고,
- 꽉꽉 대면
- 오리

정적 타입 언어의 비용

1. 타입 선언
2. 프로그램이 시스템에 강하게 묶이는 문제

empty Method 예제

Java 예제

```
public class Empty
{
    public boolean isEmpty(String s)
    {
        return s.length() == 0;
    }
}
```

Ruby 예제

```
def empty?(s)
  s.length == 0
end
```

Ruby vs Java

- Duck Typing에서는 length가 있는 모든 클래스에서 사용 가능

Unit Test

- 테스트는 어느 언어에서건 필수
- xUnit

Setup

```
class EmptyTest < Test::Unit::TestCase
  def setup
    @empty_string = ''
    @one_char_string = 'X'
    @long_string = 'The rain in Spain'

    @empty_array = []
    @one_element_array = [1]
    @long_array = [1, 2, 3, 4, 5, 6]
  end
  ...
end
```

Test

```
class EmptyTest < Test::Unit::TestCase
  ...
  def test_empty_on_strings
    assert empty?(@empty_string)
    assert ! empty?(@one_char_string)
    assert ! empty?(@long_string)
  end

  def test_empty_on_arrays
    assert empty?(@empty_array)
    assert ! empty?(@one_element_array)
    assert ! empty?(@long_array)
  end
end
```

함정

- 하나에서부터 시작하기
 - 이 예제에서는 HTML에서 시작하기
- 모든 경우를 커버하려고 하지 말 것

Templates in the Wild

Webrick 서버

```
class HelloServer < WEBrick::GenericServer
  def run(socket)
    socket.print('Hello TCP/IP world')
  end
end

s = HelloServer.new(:Port => 2000)
trap("INT"){ s.shutdown}
s.start
```

Ruby initialize Method

Thank you