

3

클래스와 메소드 Class & Method



클래스 vs 객체

메소드

객체지향 설계

Object-Oriented Program Design

- 설계도 = 클래스 다이어그램 class diagram
- 설계도 부품 = 클래스 class = 객체object를 만들기 위한 형판template
- 클래스 별로 필요한 메소드 method 명세

MyClassName

```
<return_type> myMethodName(<type_1> par_1, ..., <type_n> par_n)
```

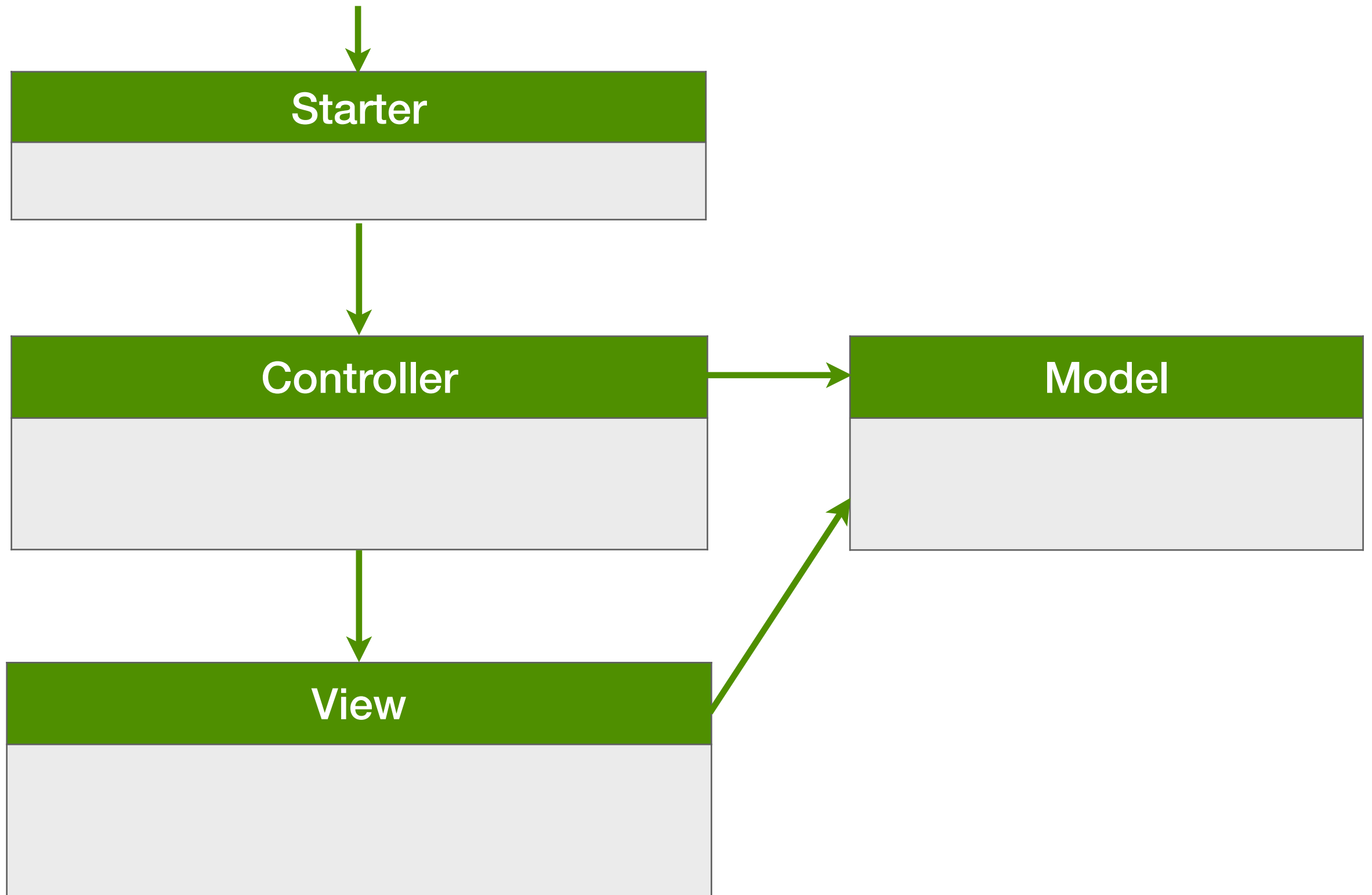
클래스의 종류

Class

구분	역할
Starter	시동 걸기
M odel	핵심 계산을 하는 두뇌 선수
V iew	외부와 소통 창구 Input / Output 으로 구분
C ontroller	매니저 지휘자

MVC (Model-View-Controller) Architecture

설계도 구조 : 클래스 다이어그램



사례 1. 요구 사항

다음과 같은 환영 메시지를 보여주세요.

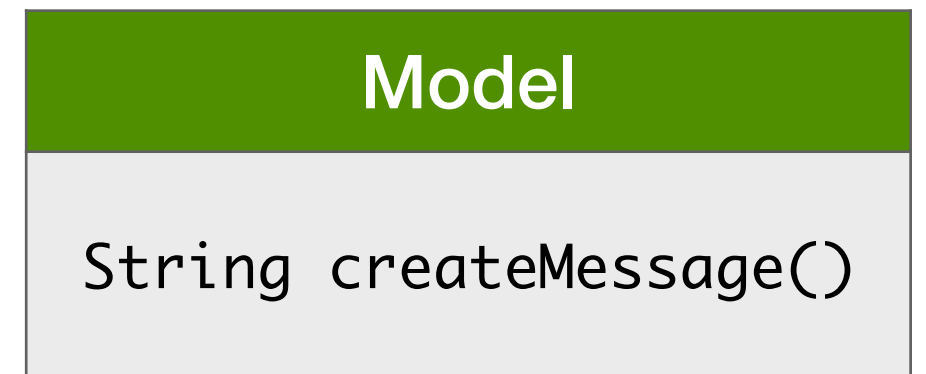
Hello, World!

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
  
}
```

```
import javax.swing.*;  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        JOptionPane.showMessageDialog(null, "Hello, World!");  
    }  
  
}
```

클래스 다이어그램

Class Diagram



클래스 다이어그램

Class Diagram

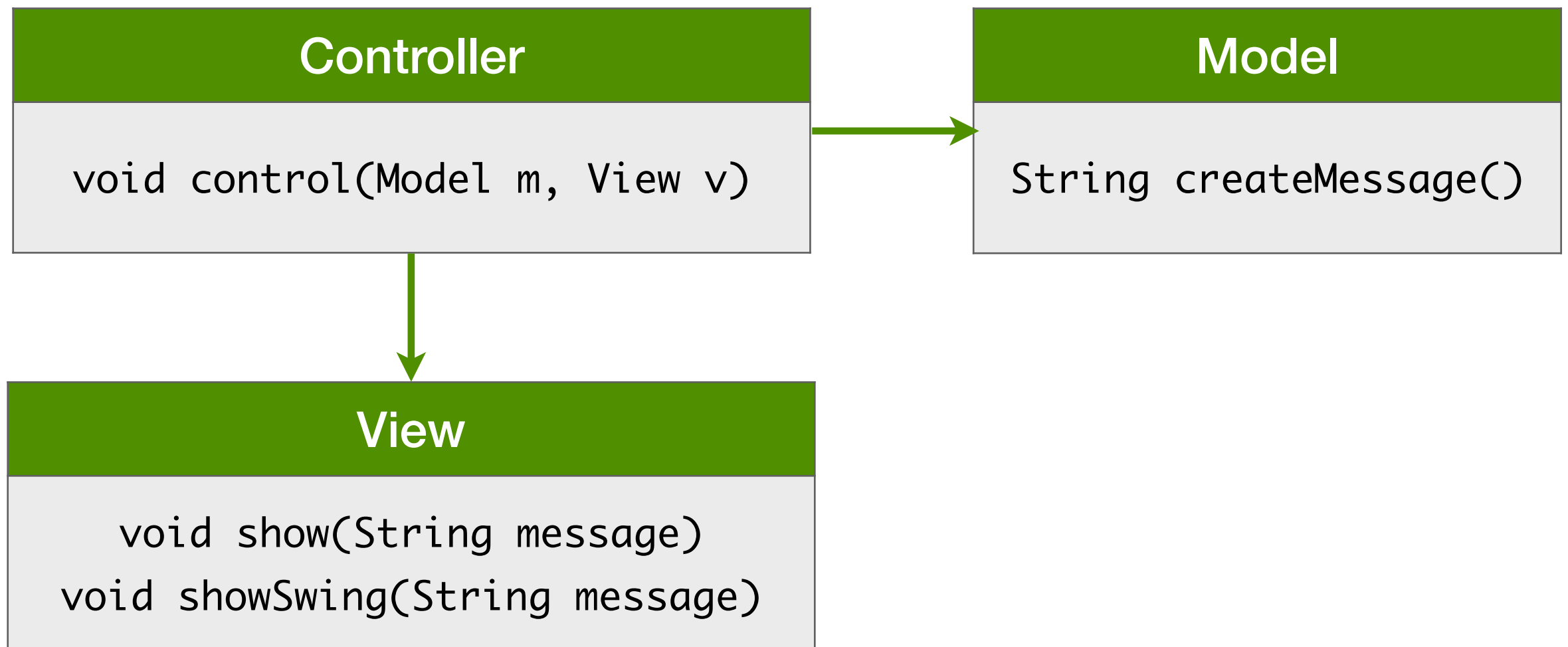
Model

String createMessage()

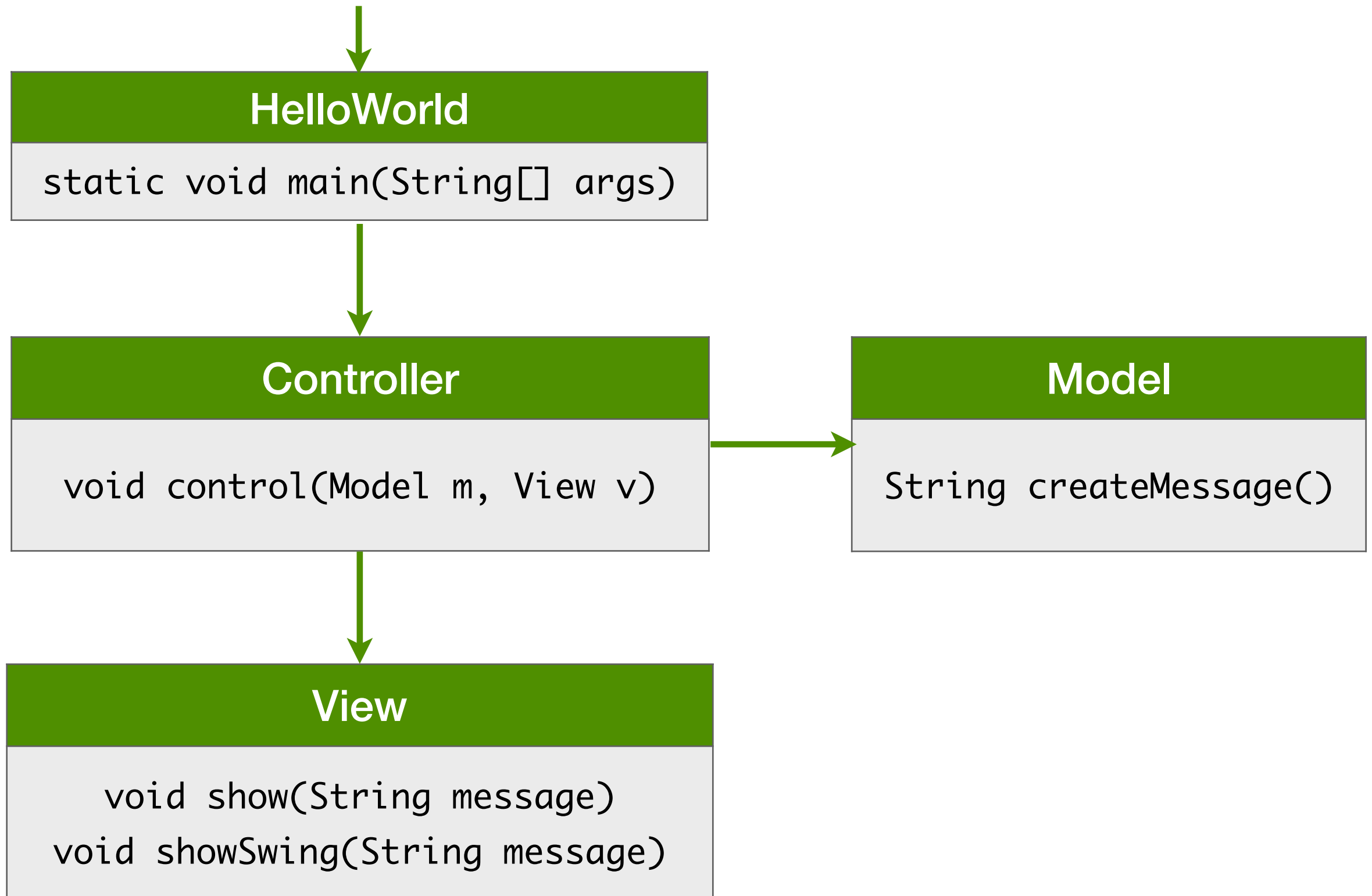
View

void show(String message)
void showSwing(String message)

클래스 다이어그램 Class Diagram



클래스 다이어그램 Class Diagram



사례 2. 요구 사항

현재 시각을 알려주세요.

```
import java.time.*;

public class Clock {

    public static void main(String[] args) {
        System.out.println(LocalTime.now());
    }

}
```

```
import java.time.*;
import javax.swing.*;

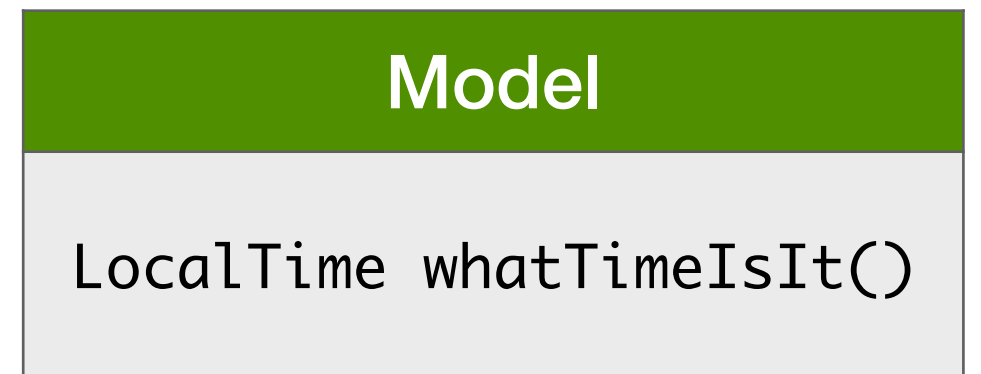
public class Clock {

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, LocalTime.now());
    }

}
```

클래스 다이어그램

Class Diagram



클래스 다이어그램

Class Diagram

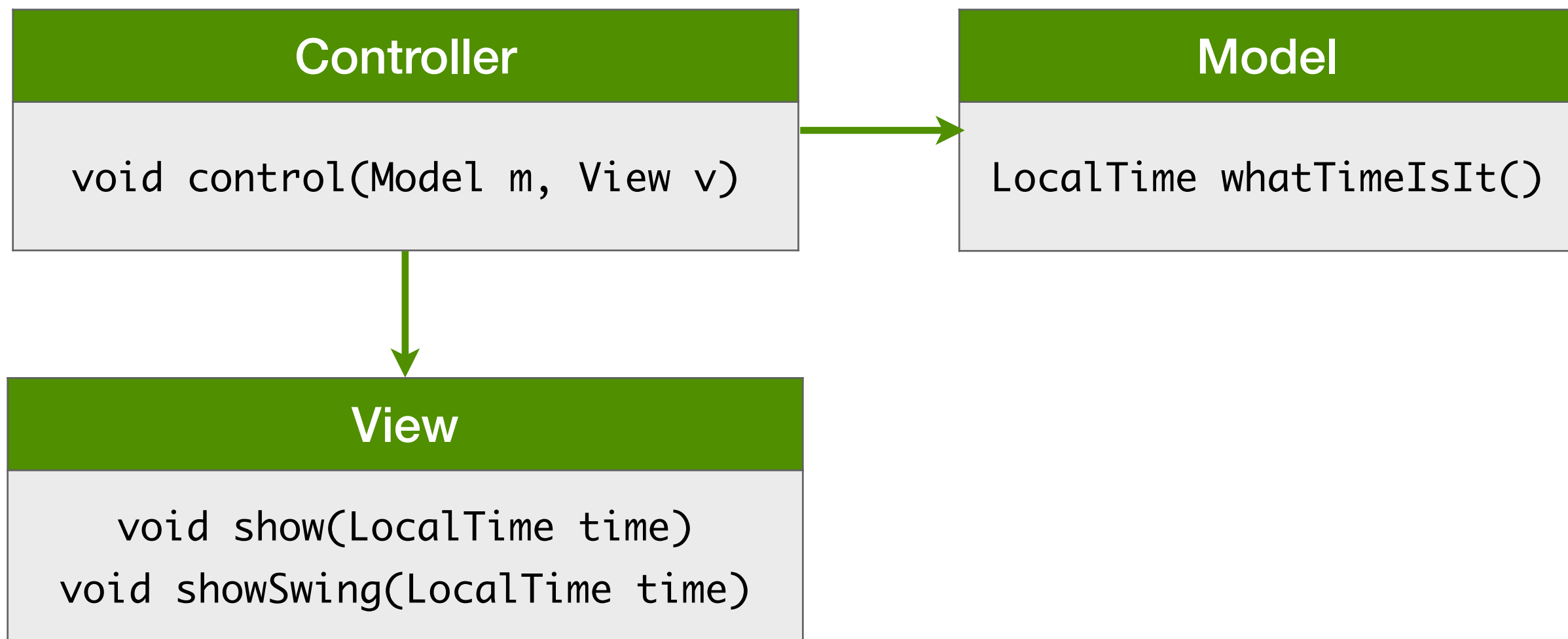
Model

```
LocalTime whatTimeIsIt()
```

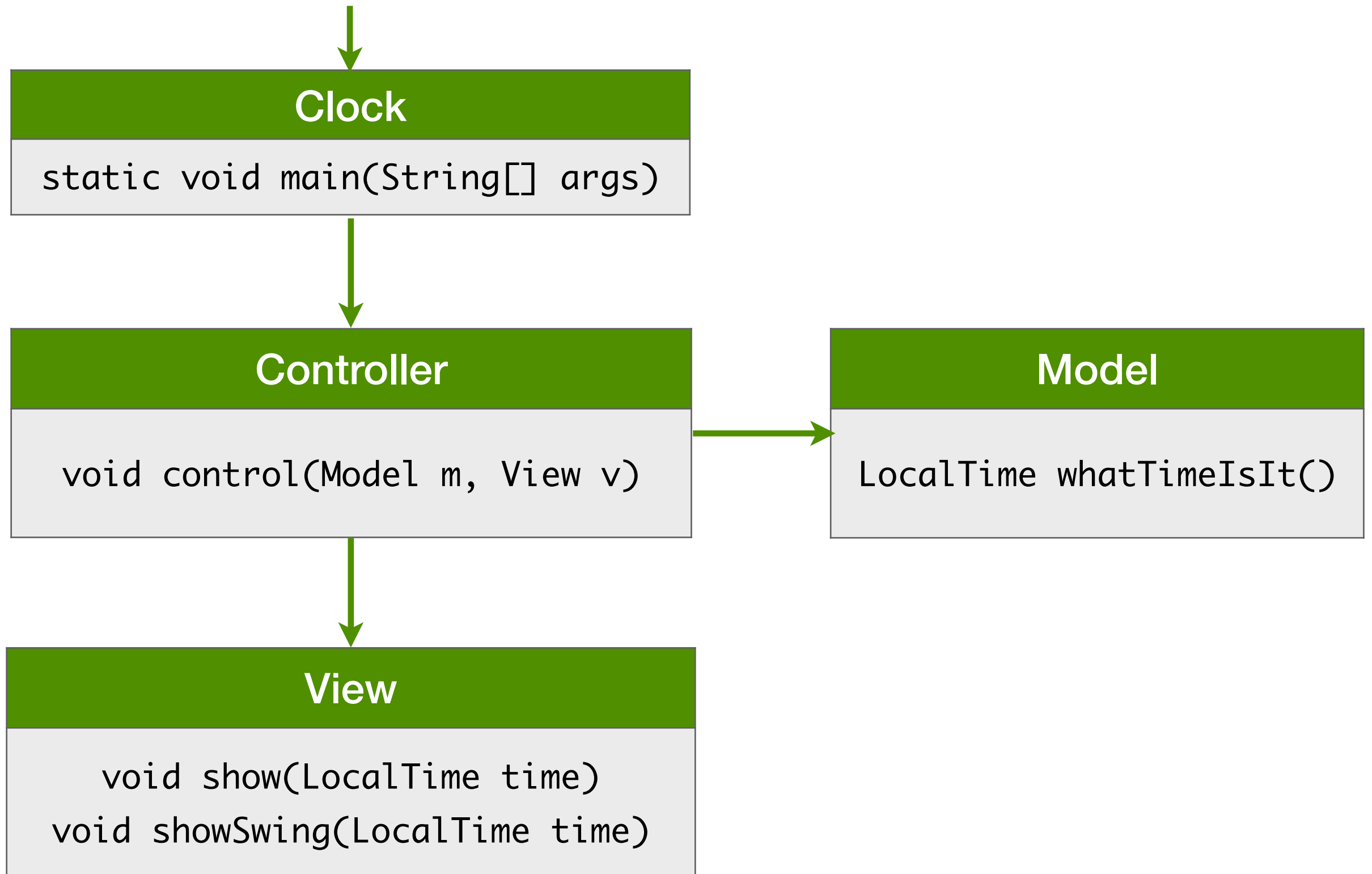
View

```
void show(LocalTime time)  
void showSwing(LocalTime time)
```

클래스 다이어그램 Class Diagram



클래스 다이어그램 Class Diagram

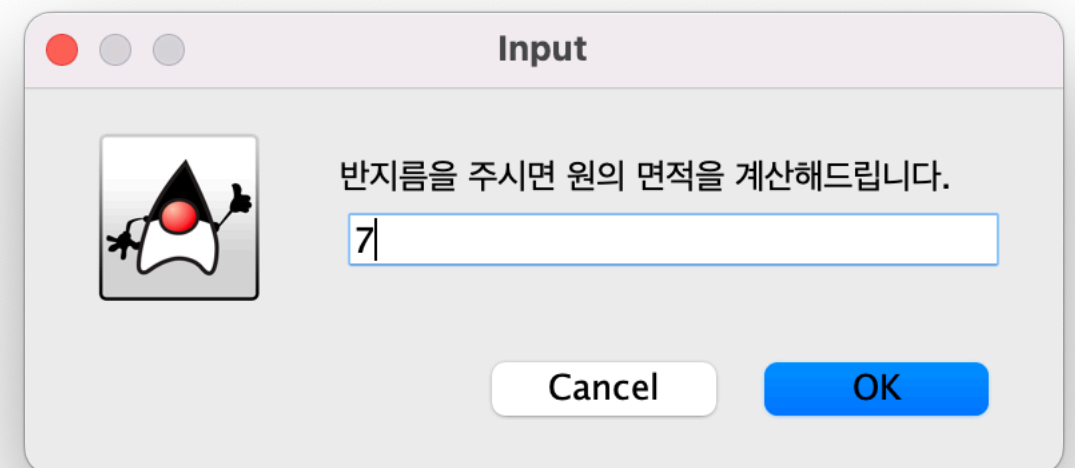
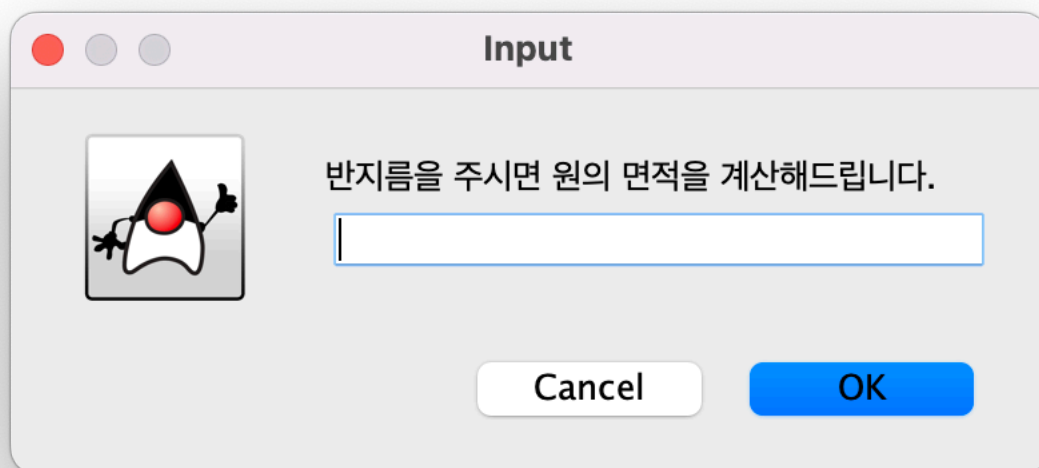


사례 3. 요구 사항

반지름을 가지고 원의 면적을 구해주세요.

```
import java.text.*;
import javax.swing.*;

public class Circle {
    public static void main(String[] args) {
        String message = "반지름을 주시면 원의 면적을 계산해드립니다.";
        String input = JOptionPane.showInputDialog(message);
        int radius = Integer.parseInt(input);
        double area = Math.PI * Math.pow(radius,2);
        DecimalFormat f = new DecimalFormat("0.00");
        System.out.print("반지름이 " + radius + "인 원의 면적은 ");
        System.out.println(f.format(area) + " 입니다.");
    }
}
```



클래스 다이어그램

Class Diagram

Model

```
double areaCircle(double radius)
```

클래스 다이어그램

Class Diagram

Model

```
double areaCircle(double radius)
```

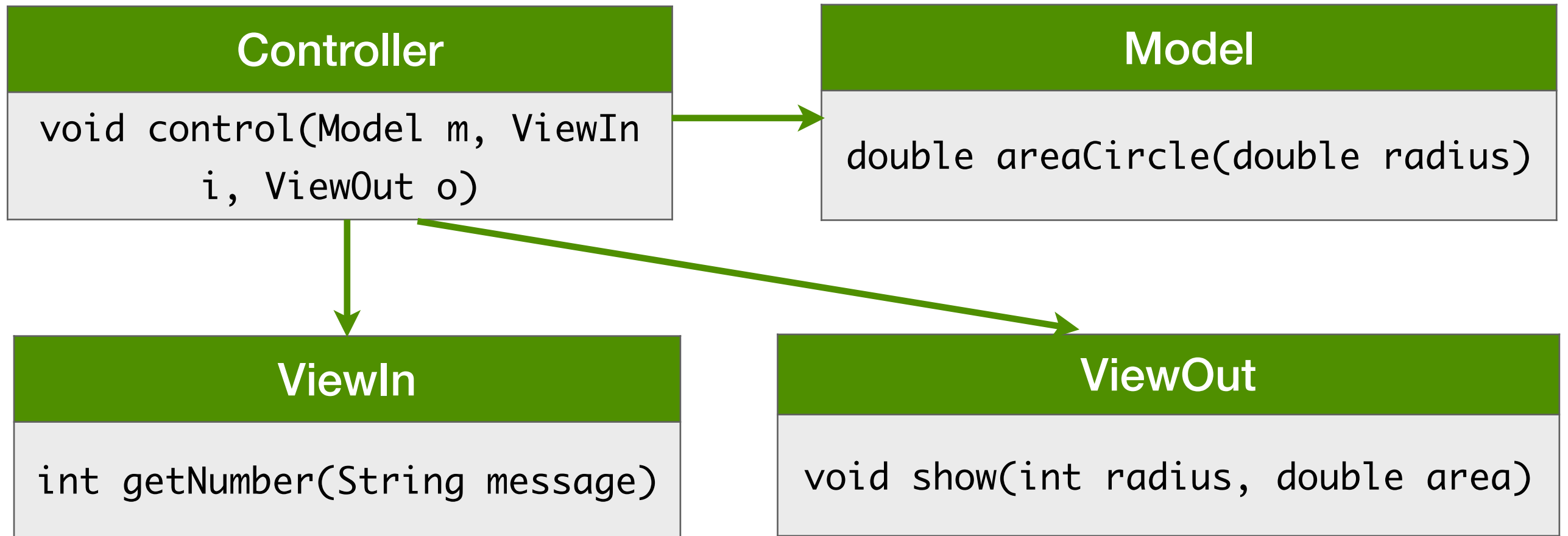
ViewIn

```
int getNumber(String message)
```

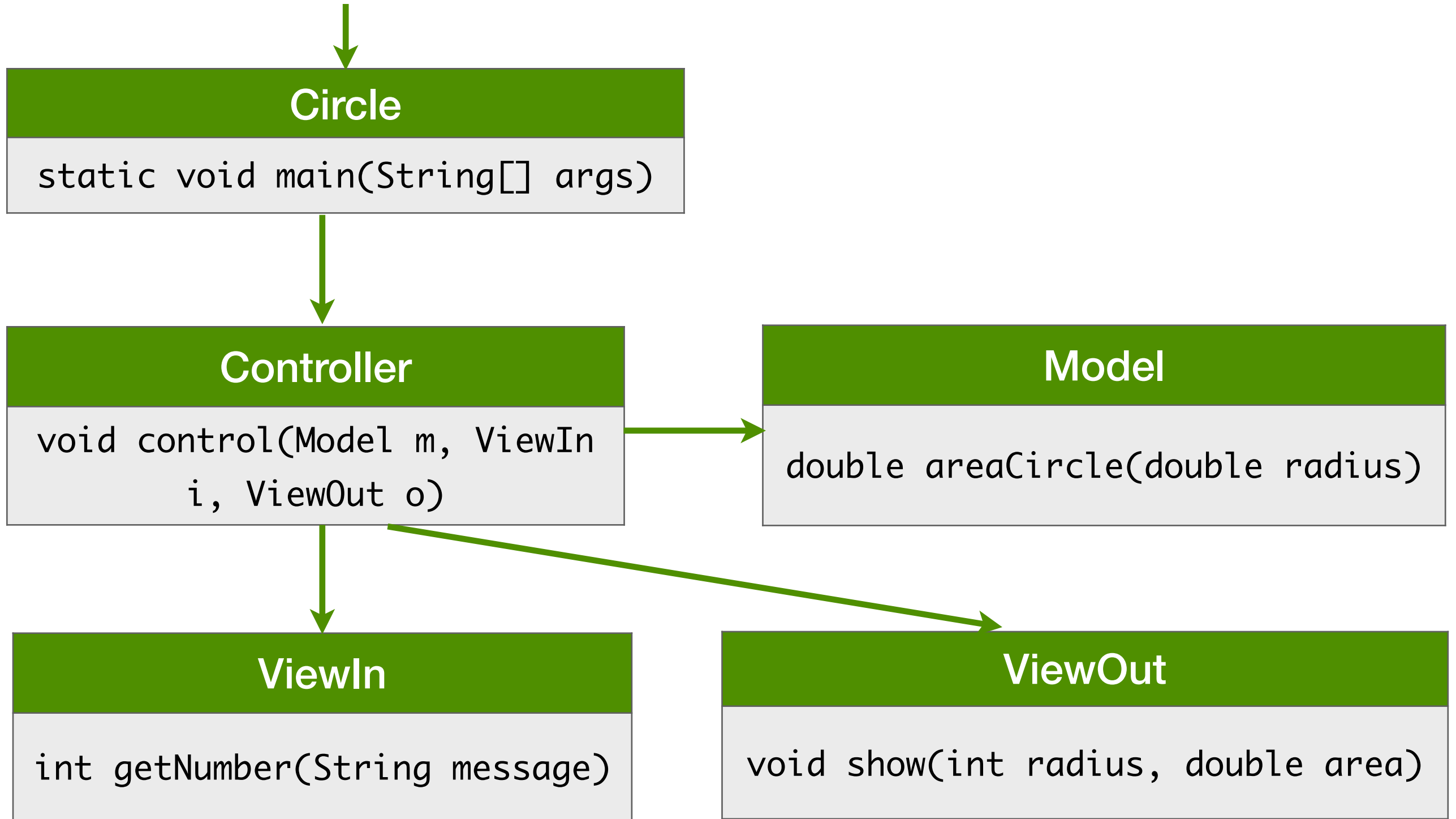
ViewOut

```
void show(int radius, double area)
```

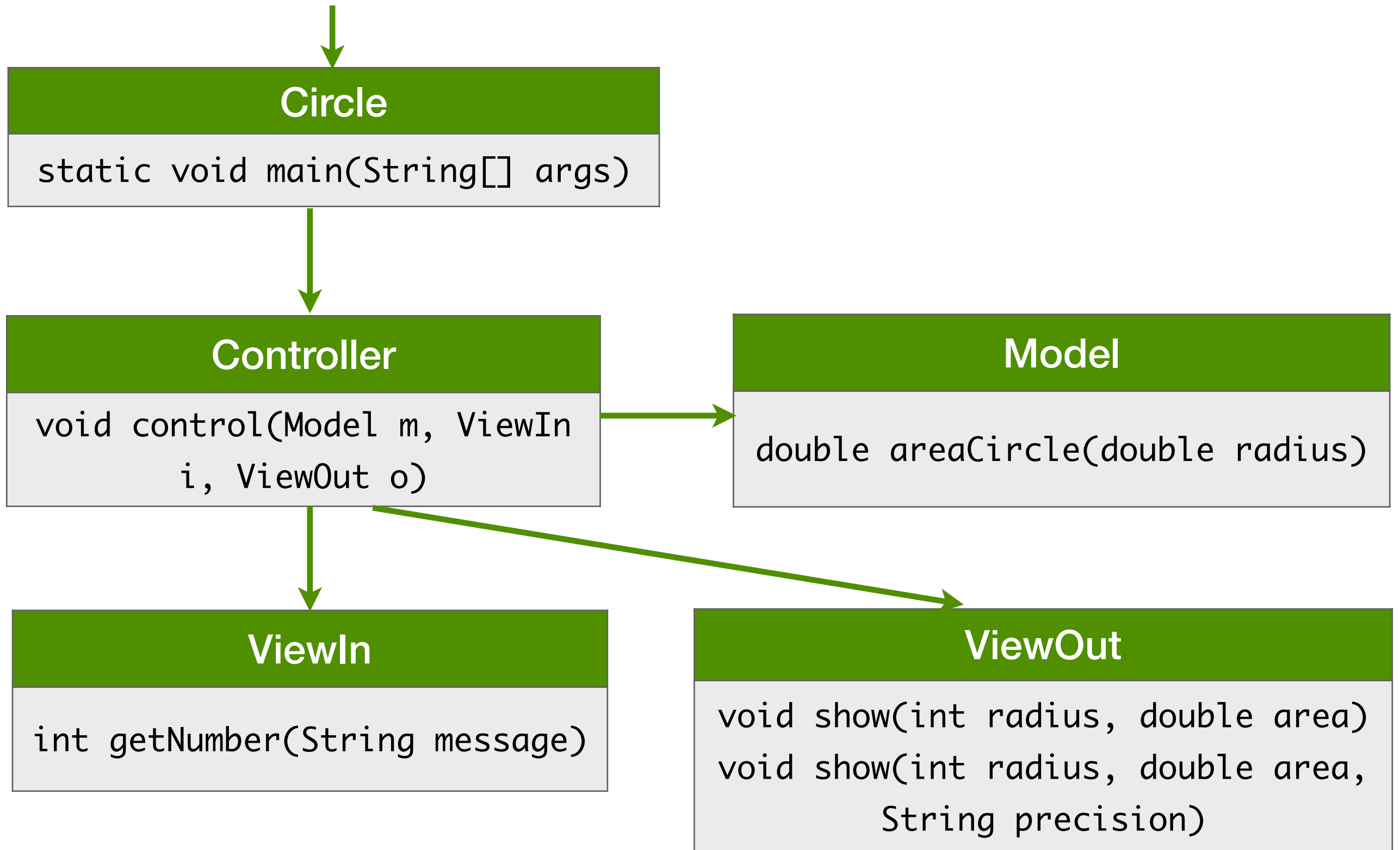
클래스 다이어그램 Class Diagram



클래스 다이어그램 Class Diagram



클래스 다이어그램 Class Diagram



실습

실습 1. 크리스마스까지 몇달 며칠 남았을까? 설계도 - 클래스 다이어그램

실습 1. 크리스마스까지 몇달 며칠 남았을까?

설계도 - 클래스 다이어그램

Model

Period countdownXmas()

실습 1. 크리스마스까지 몇달 며칠 남았을까?

설계도 - 클래스 다이어그램

Model

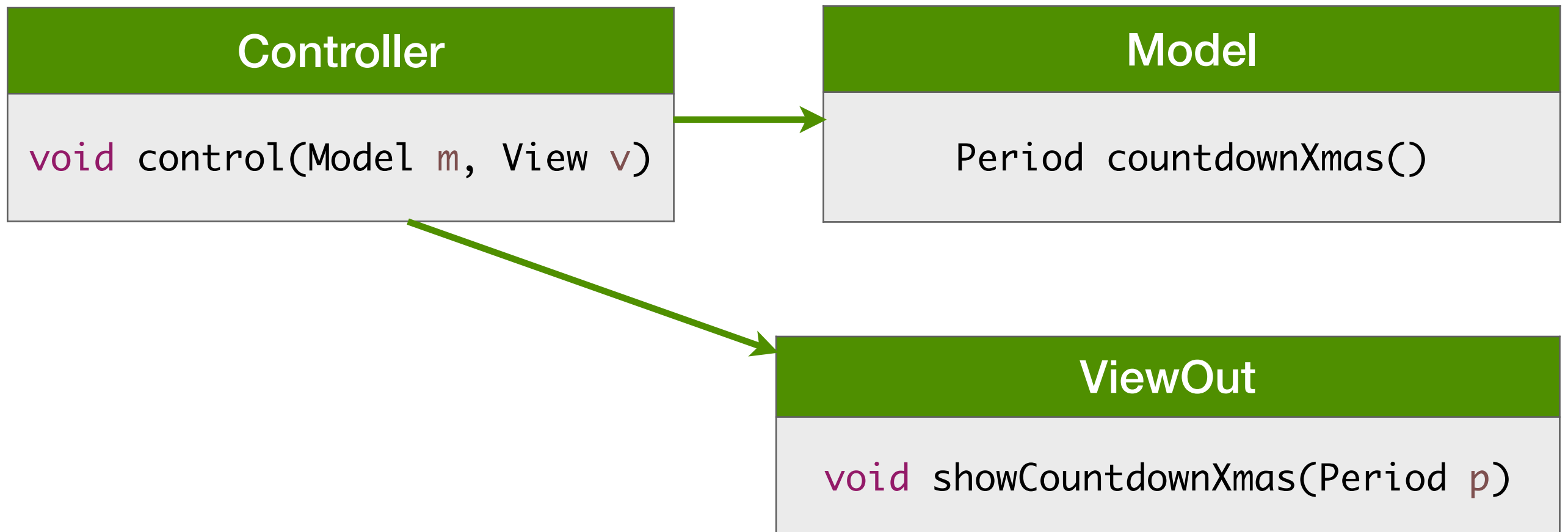
```
Period countdownXmas()
```

ViewOut

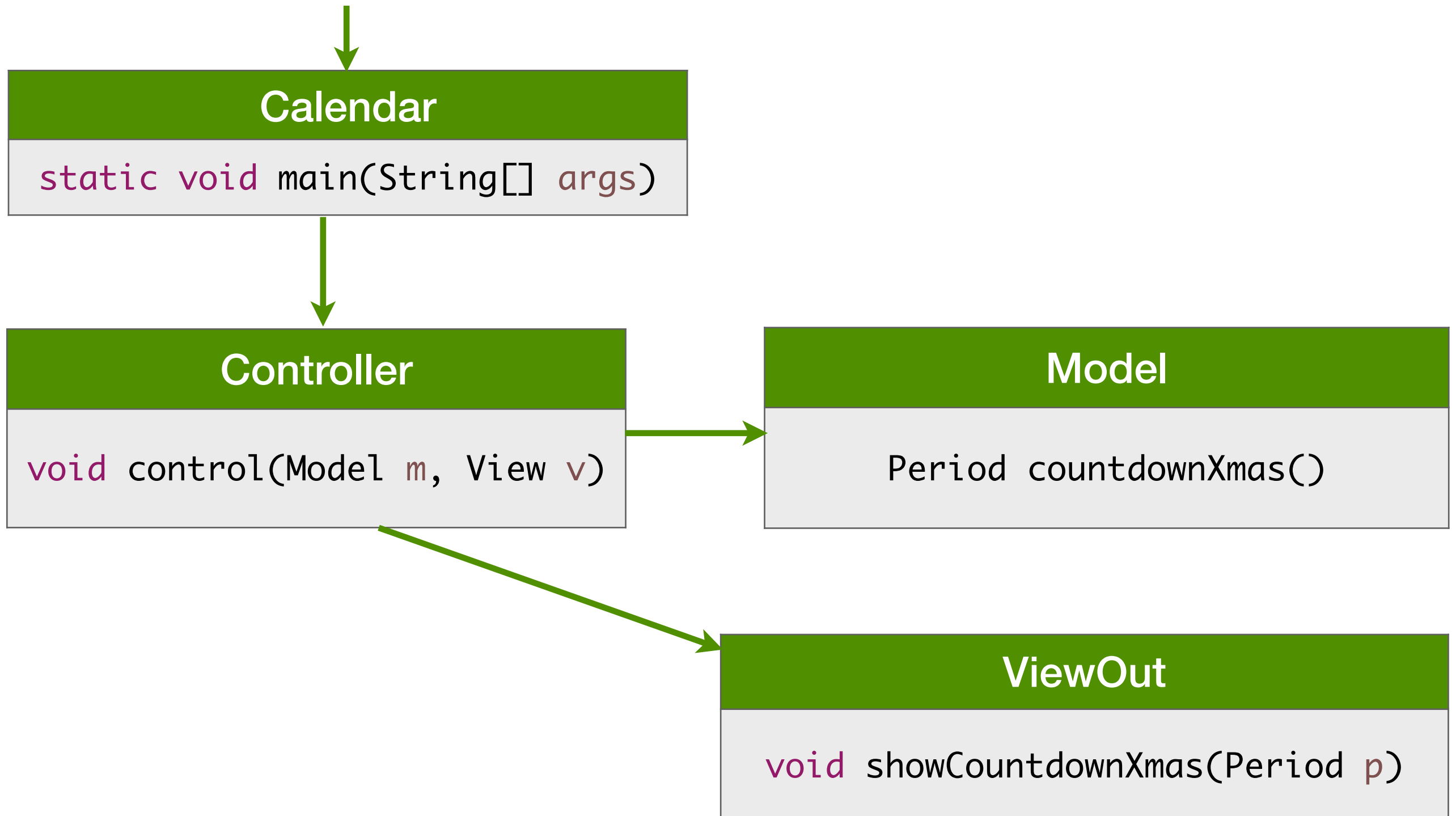
```
void showCountdownXmas(Period p)
```

실습 1. 크리스마스까지 몇달 며칠 남았을까?

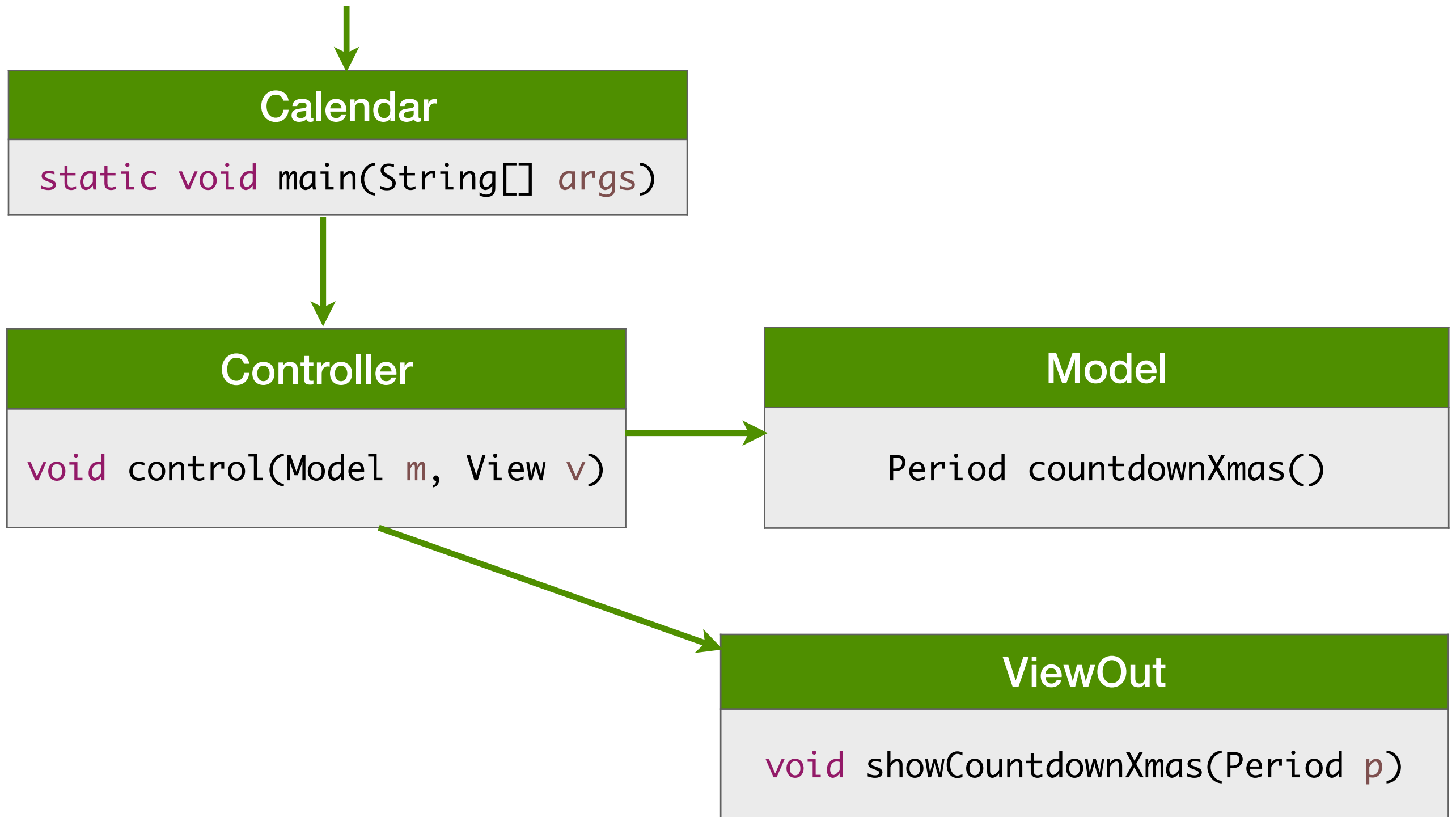
설계도 - 클래스 다이어그램



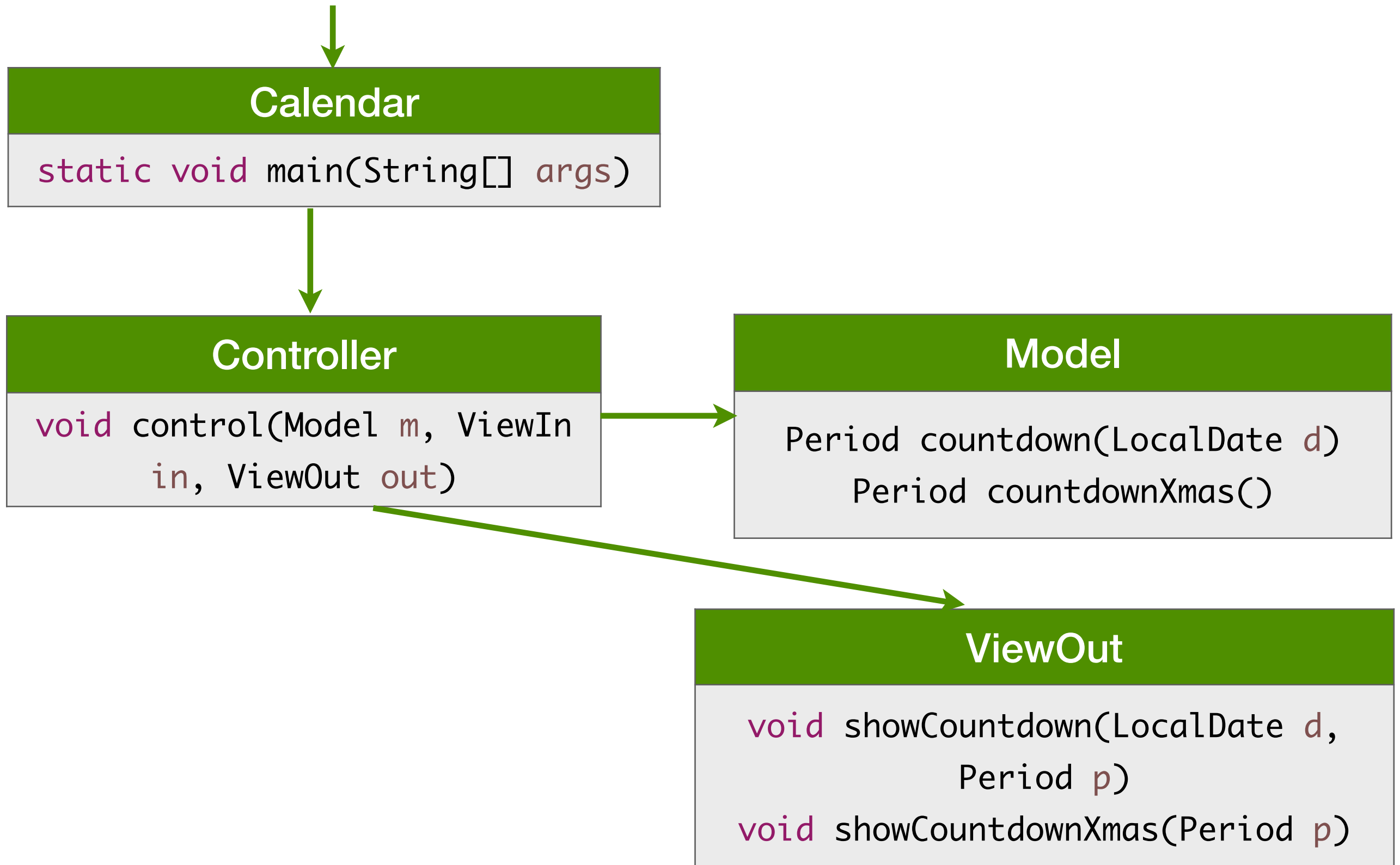
실습 1. 크리스마스까지 몇달 며칠 남았을까? 설계도 - 클래스 다이어그램



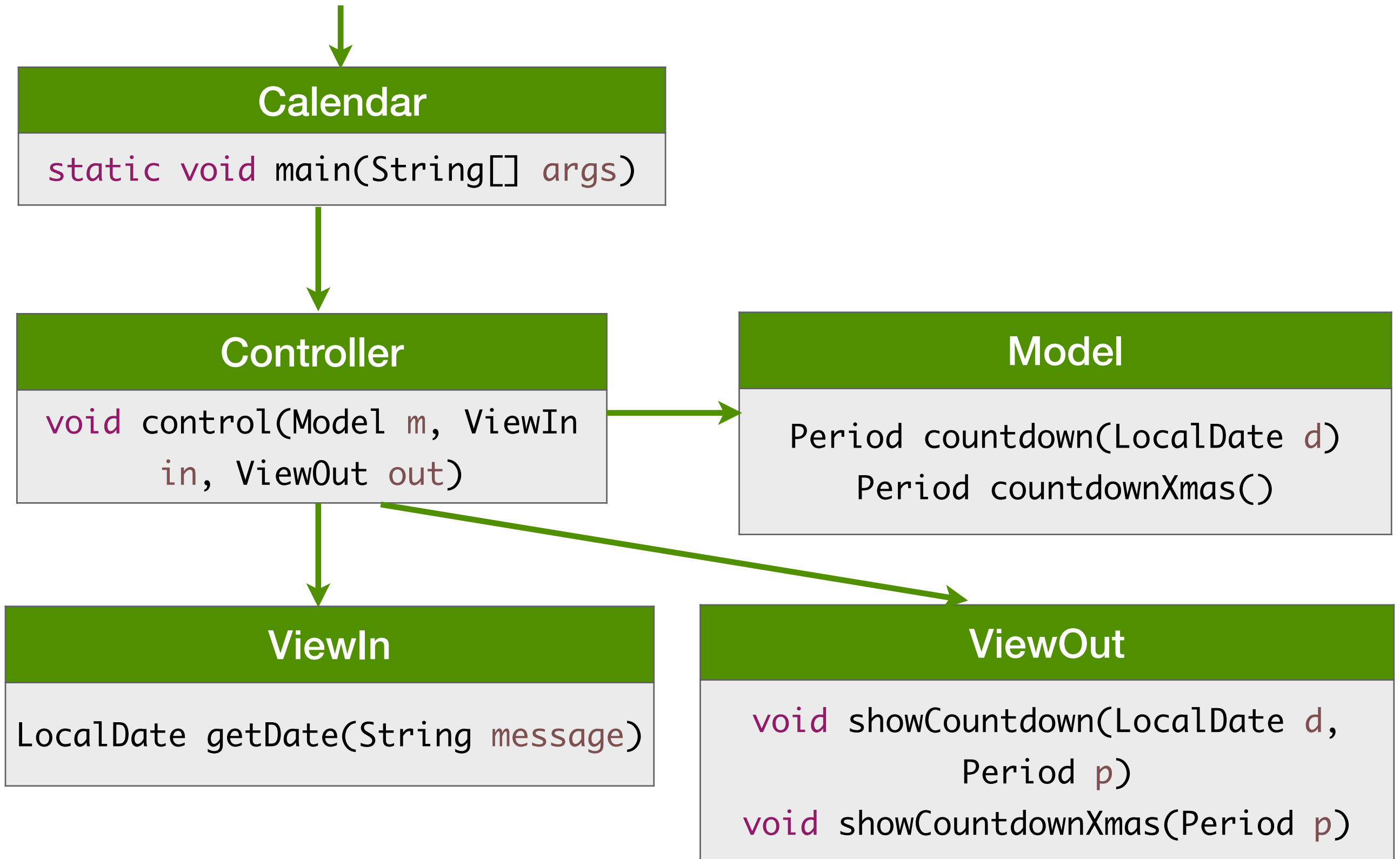
실습 2. 몇월 며칠까지 몇달 며칠 남았는지 계산 기능 추가 설계도 - 클래스 다이어그램



실습 2. 몇월 며칠까지 몇달 며칠 남았는지 계산 기능 추가 설계도 - 클래스 다이어그램



실습 2. 몇월 며칠까지 몇달 며칠 남았는지 계산 기능 추가 설계도 - 클래스 다이어그램



숙제

문제 1. 오늘부터 100일 뒤는 몇 년, 몇 월, 며칠?

Calendar

```
static void main(String[] args)
```

Controller

```
void control(Model m, ViewIn  
in, ViewOut out)
```

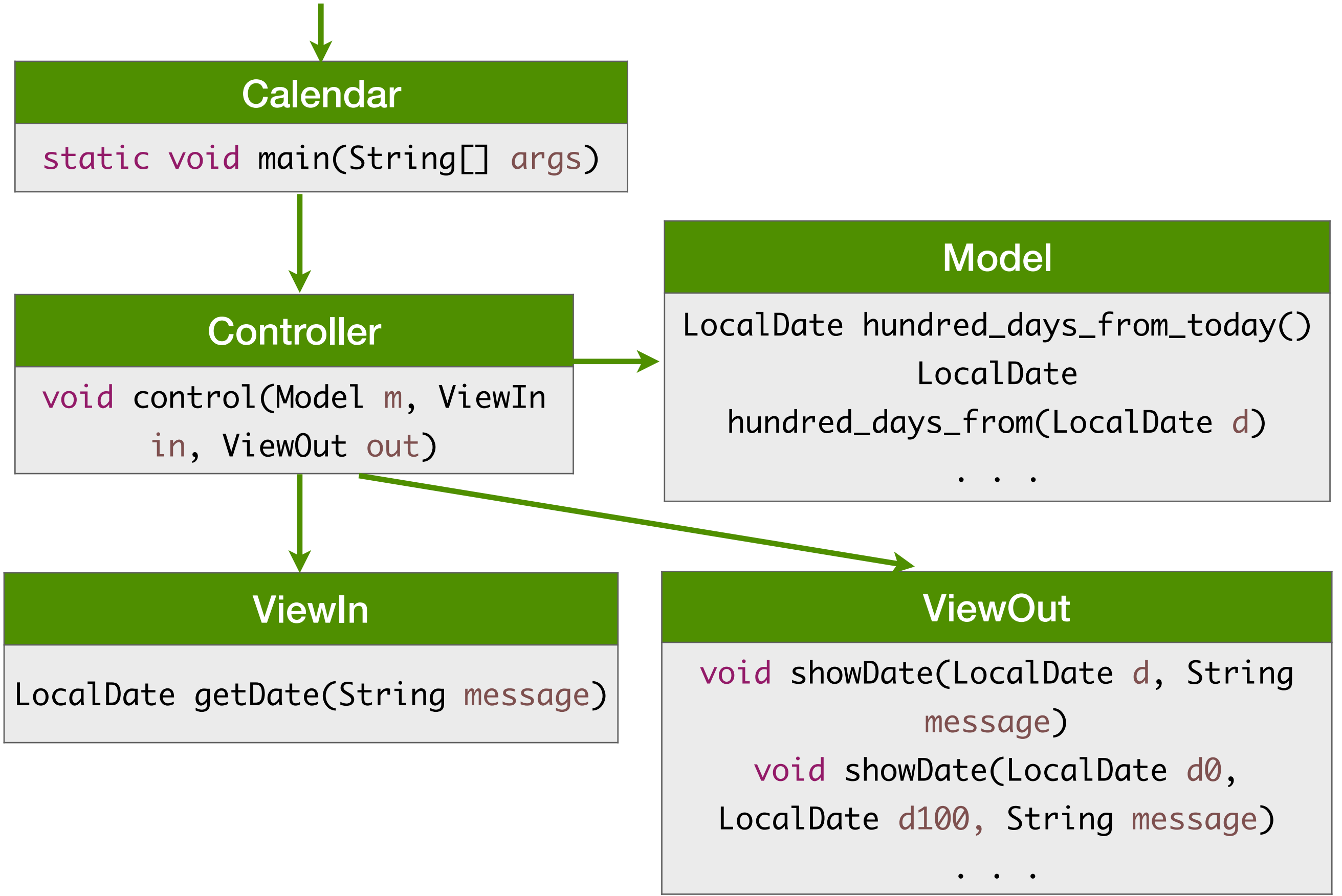
Model

```
LocalDate hundred_days_from_today()  
...
```

ViewOut

```
void showDate(LocalDate d, String  
message)  
...
```


문제 2. 특정 년,월,일부터 100일 뒤는 몇 년, 몇 월, 며칠?



Calendar

```
static void main(String[] args)
```

Controller

```
void control(Model m, ViewIn in, ViewOut out)
```

Model

```
LocalDate hundred_days_from_today()
LocalDate hundred_days_from(LocalDate d)
...
```

ViewIn

```
LocalDate getDate(String message)
```

ViewOut

```
void showDate(LocalDate d, String message)
void showDate(LocalDate d0, LocalDate d100, String message)
...
```

**문제 3. 숙제#1로 완성한 애플리케이션을
MVC 아키텍처 기반으로 설계도를 작성하고
이를 바탕으로 애플리케이션을 재작성**