자료구조: 2022년 1학기 [강의]

Abstract Data Type
Model-View-Controller

강지훈
jhkang@cnu.ac.kr
충남대학교 컴퓨터융합학부

기본 자료형
(Basic Data Type)

이론적인 관점에서의 기본 자료형

이론적인 관점에서, 자료형은 원소 (값) 들의 집합

```
int   count ;
```

- 실행적 관점:
  - 메모리에 4 byte 정수를 저장할 공간이 확보되고 그 공간의 이름이 “count” 이다.
  - 즉, 자료형 int 는 필요한 메모리를 확보하는데 필요한 정보를 제공하는 설계도 역할을 한다.
- 이론적 관점:
  - 자료형 int 는, 4 byte 로 표현되는 모든 정수들의 집합

이론적인 관점에서의 기본 자료형

- 이론적인 관점에서, 자료형은 원소 (값) 들의 집합

```
int  count ;
```

- 자료형 "int"
  - int = { -2^{31}, … , -1, 0, 1, 2, 3, … , $+(2^{31}-1)$ }
  - 변수 선언의 의미:
    - 변수 count 는 집합 int 중에서 한 개의 원소 (값)을 가질 수 있다.

이론적인 관점에서의 기본 자료형

- 이론적인 관점에서, 자료형은 원소 (값) 들의 집합

```
int  count ;
```

- 자료형 "int"
  - int = { -2^{31}, … , -1, 0, 1, 2, 3, … , $+(2^{31}-1)$ }

- 자료형 "boolean"
  - boolean = { false, true }
  - boolean found ;
    - 변수 found 는 집합 boolean 의 원소 중 하나의 값을 갖는다. 즉, found 는 false 의 값이거나 true 의 값을 갖는다.

이론적인 관점에서의 기본 자료형

- 이론적인 관점에서, 자료형은 원소 (값) 들의 집합

- Java 에서의 활용:
  - Enumeration Data Type (열거 자료형)

자료형을 사용하는 이유는?

- 자료의 형을 명시적으로 선언하면 비명시적인 경우에 비해 사용자가 잘못 사용할 가능성을 줄여줄 수 있다.
  - 컴파일러는 프로그래머가 잘못 사용하여 자료형이 일치하지 않은 것을 찾아낸다.

- 변수의 자료형을 명시적으로 선언하면, 다른 사용자에게 그 변수의 용도를 이해하는데 도움을 줄 수 있다.

Java 에서의 기본 자료형

- Java 에서 사용할 수 있는 기본 자료형
  - boolean, byte, char
  - short, int, long, float, double
- Java를 포함한 대부분의 프로그램 언어는 최소한의 기본자료형을 제공한다.
- 더 복잡한 자료형이 필요하다면?
  - 배열
  - Class (추상자료형)

기본 자료형과 관련 연산

- 자료형은 관련된 연산을 가지고 있다.
  - 그 자료형의 값에 적용할 수 있는 행위가 있다.
  - 자료형마다 관련된 연산들은 동일하지는 않다.
- 예:
  - int: +, -, *, /, %, ++, --, ...
  - double: +, -, *, /, ...
    - 실수(double) 자료형은 %, ++, - - 과 같은 연산은 가지고 있지 않다.
- 기본자료형은 관련된 연산을 묵시적으로 가지고 있다.
- 모든 자료형은 관련된 연산을 가지고 있다.
  - Class (추상자료형)의 관련 연산은 명시적으로 선언

추상 자료형
(Abstract Data Type)

추상 자료형 (ADT)

- 추상자료형도 자료형

- 객체 인스턴스에 적용되는 관련 연산들이 있다.
  - 객체 인스턴스: 객체의 실체

추상화란?

- 가루로 된 매우 쓴 마이신 약을 캡슐로 감싸자!
  - 마이신은 염증 치료에 사용되는 매우 쓴 가루약
  - 염증 환자: 사용자 (User)
    - 치료를 위해 마이신 가루약을 먹는다.
    - 그렇지만 먹어서 낫기만 하면 되는 것이 목적이지, 그 약의 쓴 맛을 보는 것은 목적이 아니다.
  - 제약회사: 구현자 (Implementer)
    - 마이신 약의 목적은 살리되, 환자가 원하지 않는 약의 쓴 맛은 모르게 하고 싶다.
    - 캡슐화 (encapsulation): 약의 효능에는 영향을 주지 않으면서도 맛은 쓰지 않은 캡슐(capsule) 을 만들어 그것으로 마이신 약을 감싸서 알약으로 만든다.
    - 즉, 캡슐화된 알약을 먹으면 염증치료가 된다

캡슐화가 추상화!

- 추상화 (abstraction): 어떻게 만들어졌는지와는 관계없이, 염증치료가 되도록 가루약을 알약으로 캡슐화 하는 행위
- 사용자는 구현 방법에 독립적 (implementation-independent):
  - 사용자는 치료를 위해 단지 캡슐로 된 염증치료 알약을 먹으면 된다. 이미 효능이 보장된 캡슐 알약 속에 무엇이 들어 있는지 또는 어떻게 만들었는지는 관심을 가질 필요가 없다.
  - 구현자는 기술이 발달하여 최신기법의 염증치료 약제를 개발하게 되면 단지 캡슐 안의 성분만 바꾸면 된다.

TV 에서의 추상화

- TV의 사용자와 생산자
  - TV를 시청하려는 사용자:
    - TV를 만드는 방법에는 별 관심이 없다.
    - 단지 리모컨의 전원 단추를 누르고, 채널을 선택하고 볼륨을 조절할 수 있으면 된다.
  - TV 생산자:
    - 효율적이면서도 편리하게 사용할 수 있는 구체적인 제조 방법을 알아야만 한다.

- 리모컨과 함께 Capsule화 된 TV
  - 사용자는 TV 내부는 몰라도 리모컨 만으로 시청이 가능
  - 캡슐화가 됨으로써, TV는 누구나 편리하게 사용할 수 있도록 추상화 되었다!

객체를 사용하려면...

- 사용자에게 좋은 리모컨이 제공되어야 한다
  - 리모컨은 바로 객체의 공개함수
  - 객체에게 필요하면서도 사용하기 편리한 공개함수는?

- 구현자는 리모컨의 각 단추가 눌려졌을 때 객체가 무슨 일을 해야 할지를 정확하고 효율적으로 구현해야 한다.

추상자료형 (Abstract Data Types)

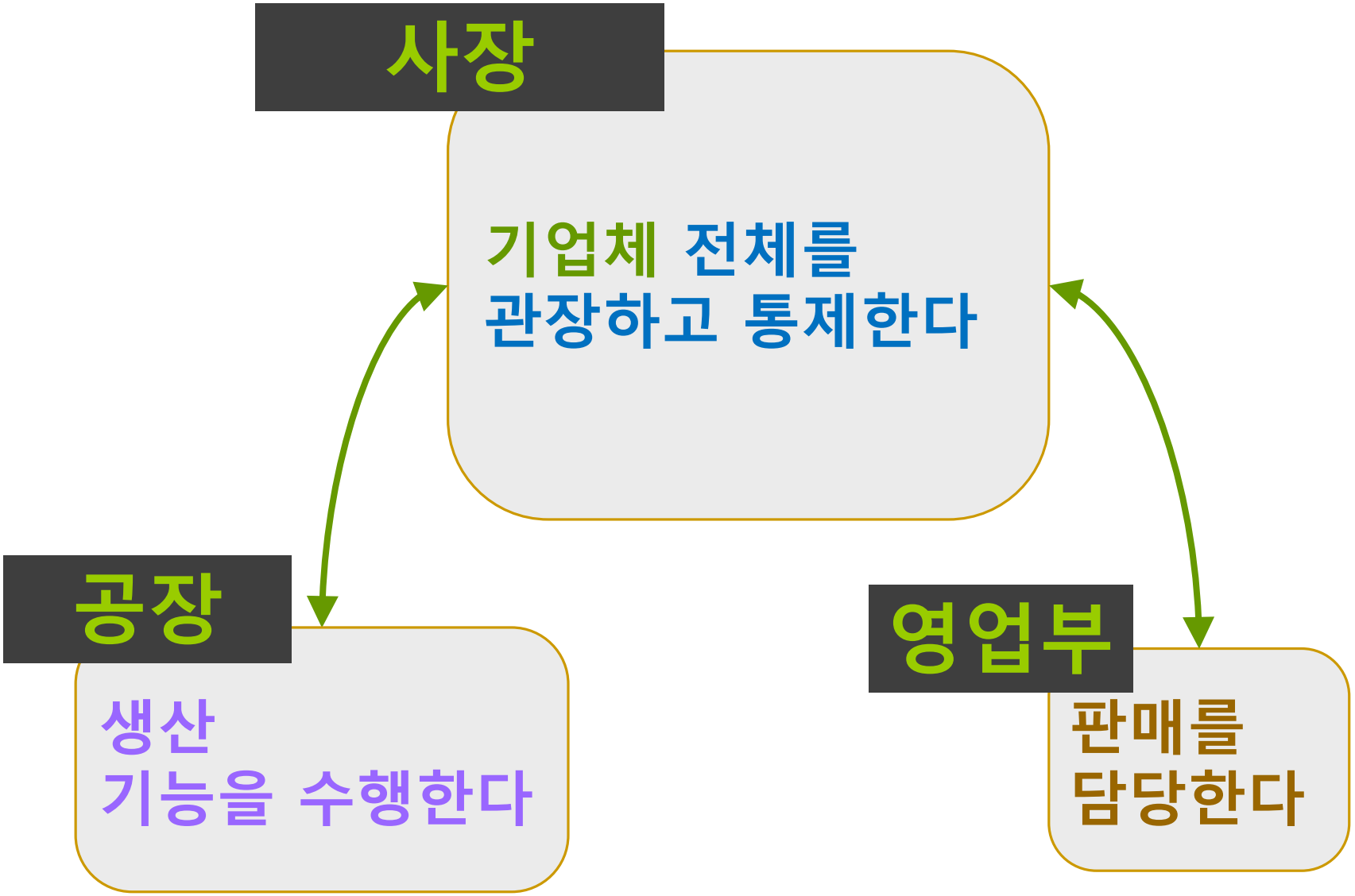
- 자료형
- 객체의 규격
- 객체에 대한 연산의 규격
- 객체의 표현과 구현의 분리
  - 구현에 독립적 (implementation-independent)

추상자료형 (Abstract Data Types)

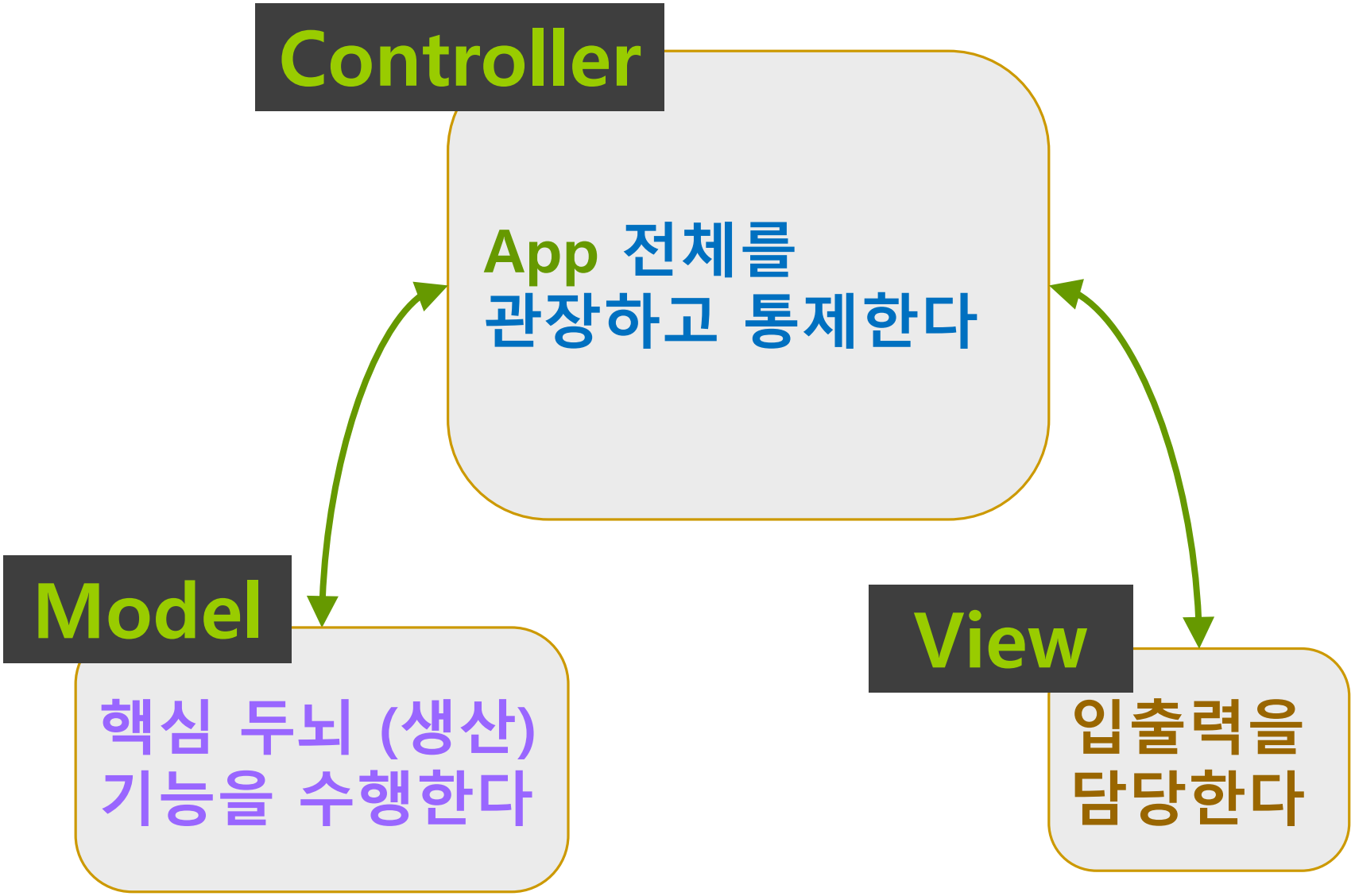
- 객체는 저장 공간이며, 또한 값
- 추상자료형의 연산
  - 생성 (Construction)
  - 소멸 (Destruction)
  - 상태 알아보기 (Observation / Reporting)
  - 상태 바꾸기 (Modification / Transformation)

Model-View-Controller

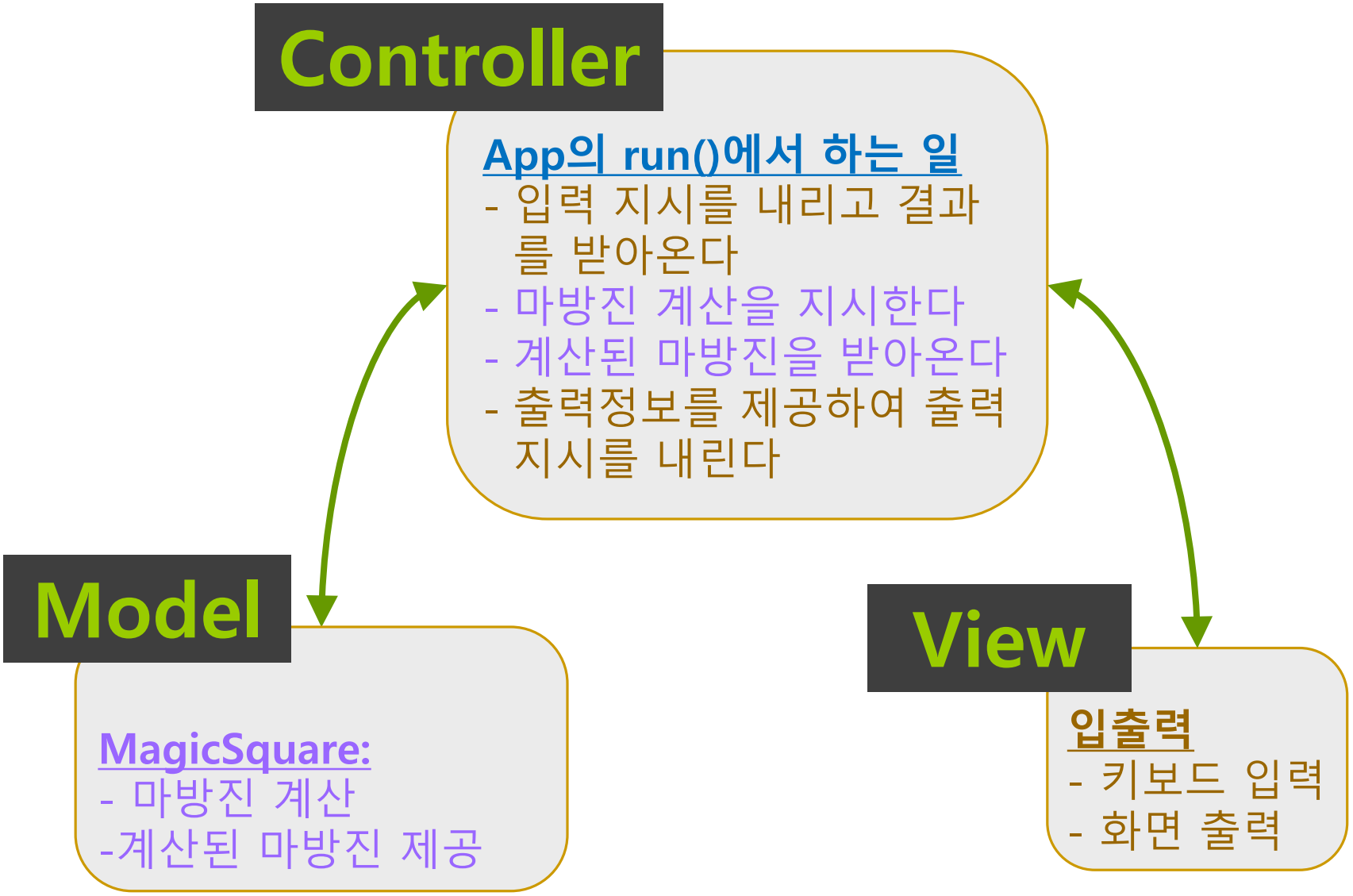
❑ App 에서의 역할 구분



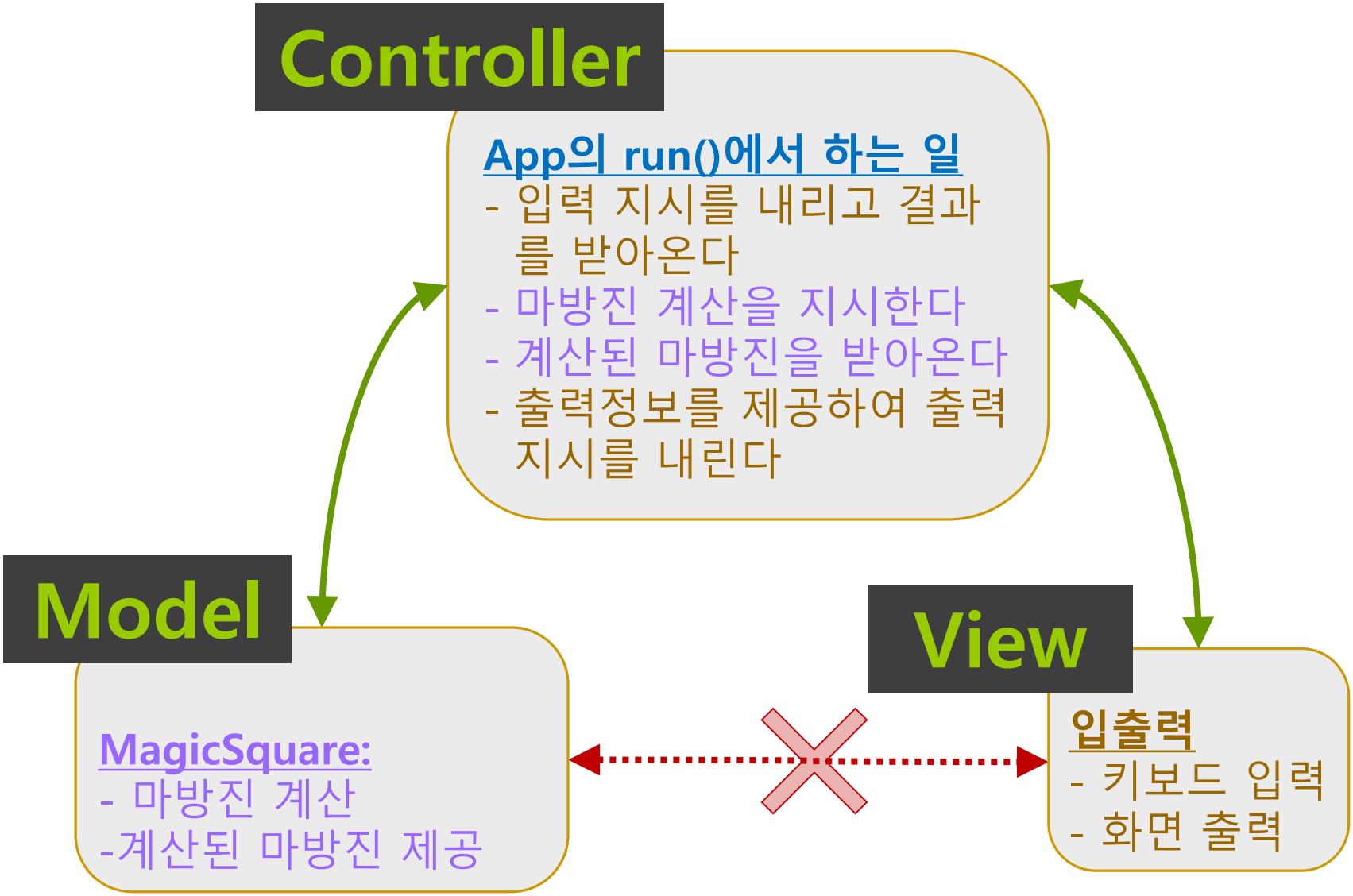
❑ App 에서의 역할 구분


Controller
App 전체를
관장하고 통제한다
Model
핵심 두뇌 (생산)
기능을 수행한다
View
입출력을
담당한다

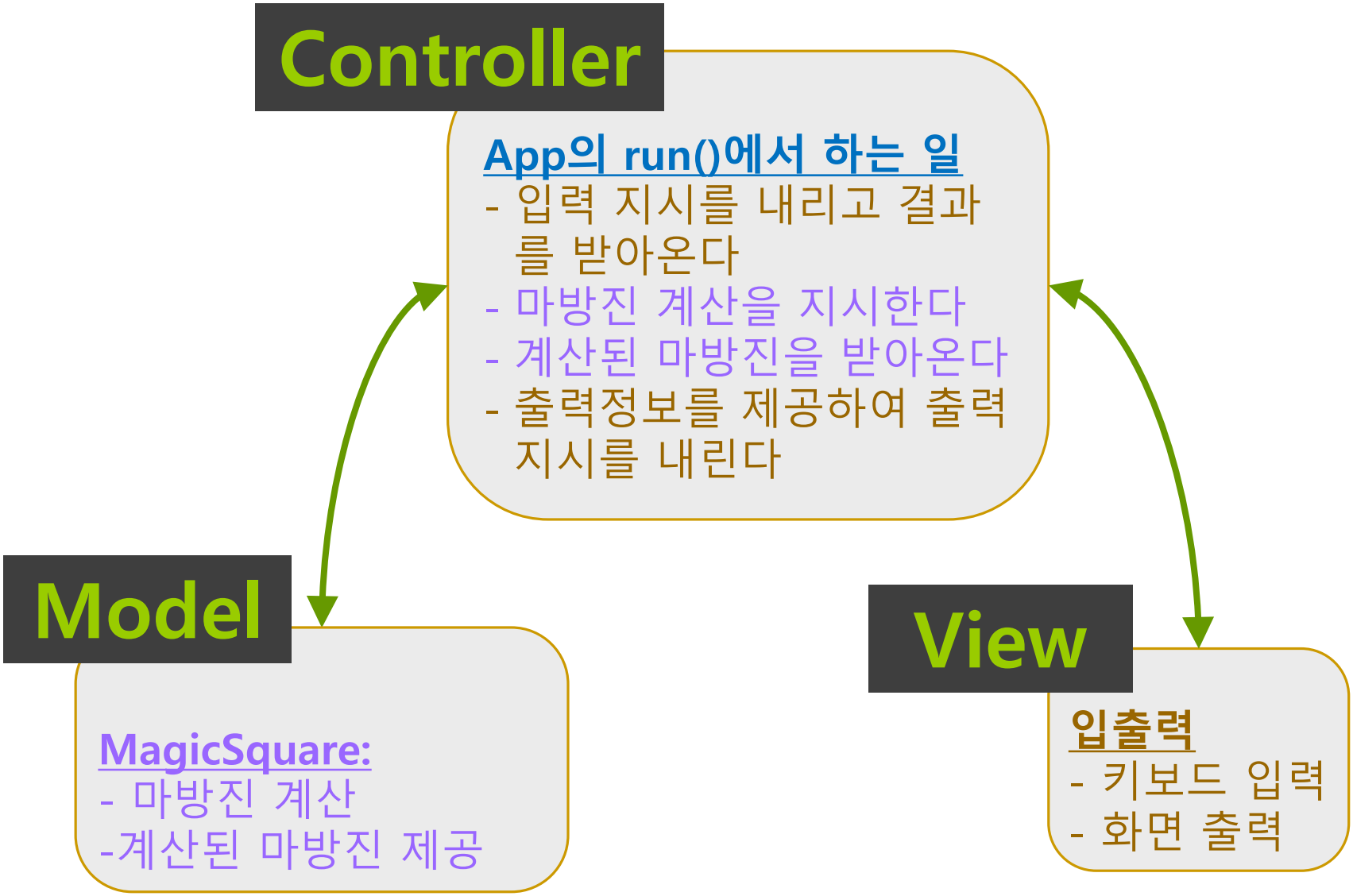
예: 마방진


Controller
App의 run()에서 하는 일
- 입력 지시를 내리고 결과를 받아온다
- 마방진 계산을 지시한다
- 계산된 마방진을 받아온다
- 출력정보를 제공하여 출력 지시를 내린다
Model
MagicSquare:
- 마방진 계산
-계산된 마방진 제공
View
입출력
- 키보드 입력
- 화면 출력

부하 직원이 상사 몰래?


Controller
App의 run()에서 하는 일
- 입력 지시를 내리고 결과를 받아온다
- 마방진 계산을 지시한다
- 계산된 마방진을 받아온다
- 출력정보를 제공하여 출력 지시를 내린다
Model
MagicSquare:
- 마방진 계산
-계산된 마방진 제공
View
입출력
- 키보드 입력
- 화면 출력

MVC


Controller
App의 run()에서 하는 일
- 입력 지시를 내리고 결과를 받아온다
- 마방진 계산을 지시한다
- 계산된 마방진을 받아온다
- 출력정보를 제공하여 출력 지시를 내린다
Model
MagicSquare:
- 마방진 계산
-계산된 마방진 제공
View
입출력
- 키보드 입력
- 화면 출력

각각을 class 로!

- 마방진 프로그램
  - Controller:
    Class "AppController"
  - Model:
    Class "MagicSquare"
  - View:
    Class "AppView"

End of

"ADT, MVC"