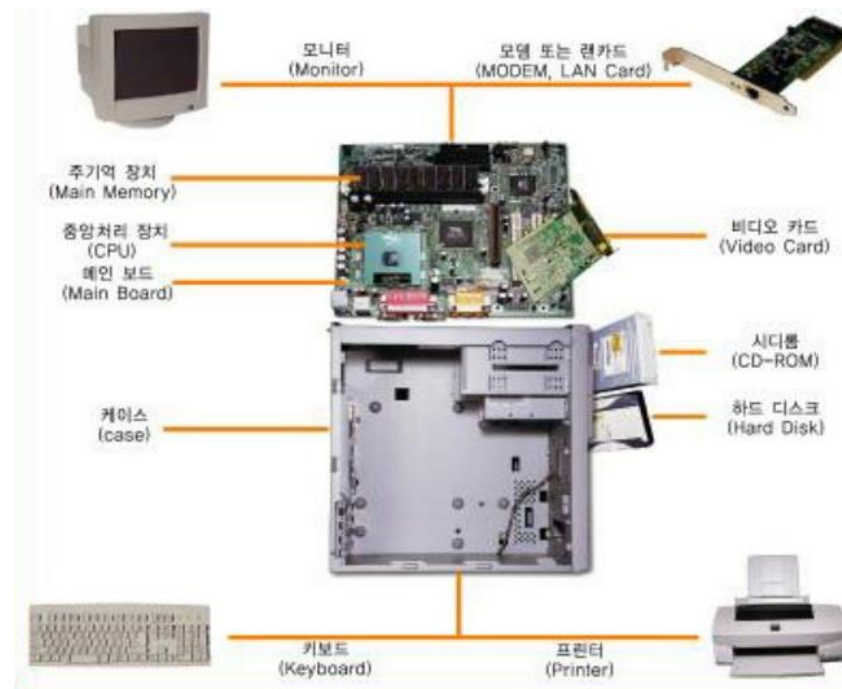


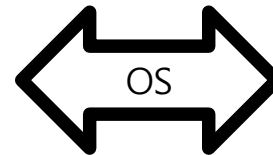
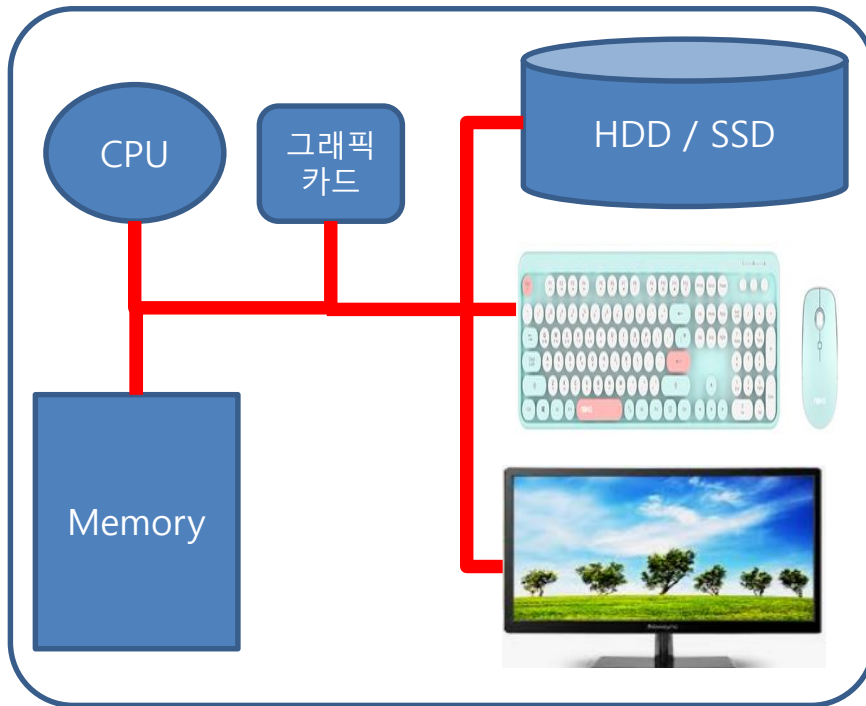
보강

2021. 05. 21

■ 컴퓨터 HW



■ HW vs. OS vs. SW



컴퓨터 사용자가 원하는 작업을 수행하는 SW를 OS위에 실행시킨다.

■ SW

✓ HW

- CPU, Memory(RAM), HDD/SDD
- HW의 가동을 돕는 드라이버는 SW임

✓ OS

- Windows, Linux(CentOS, 우분투, 레드햇)

✓ SW

- 화면구현, 프로그래밍, DBMS

패턴을 통한 객체지향 프로그램의 이해



패턴이란

우리의 개발자 선배님들이 객체지향 언어의
장점들을 모아 가장 효율적으로 개발을 할 수
있게 만들어 놓은 틀

Strategy Pattern

알고리즘 군을 정의하고 각각의
기능을 부품처럼 캡슐화하여 교환
해서 사용할 수 있도록 하는 패턴

시 나 리 오

- 모든 로봇은 기본적으로 걷고, 달릴 수 있어야 합니다.
- 로봇 모양은 팔, 다리, 머리, 몸통으로 이루어져 있다

	날 수 있을지	미사일 쏘지	검
SuperRobot	O	O	레이저검
StandardRobot	X	O	목검
LowRobot	X	X	없음

- 위 3가지 형태의 로봇을 만들되, 추후 다른 로봇을 만들어야 할 경우를 생각하여 유연한 프로그램을 만들어 보도록 합니다.

1단계. 각각 클래스 정의

SuperRobot

```
Shape(){}  
actionWalk(){  
    걸을수있다}  
actionRun(){  
    뛸수있다}  
actionFly(){O}  
actionMissile(){O}  
actionKnife(){L}
```

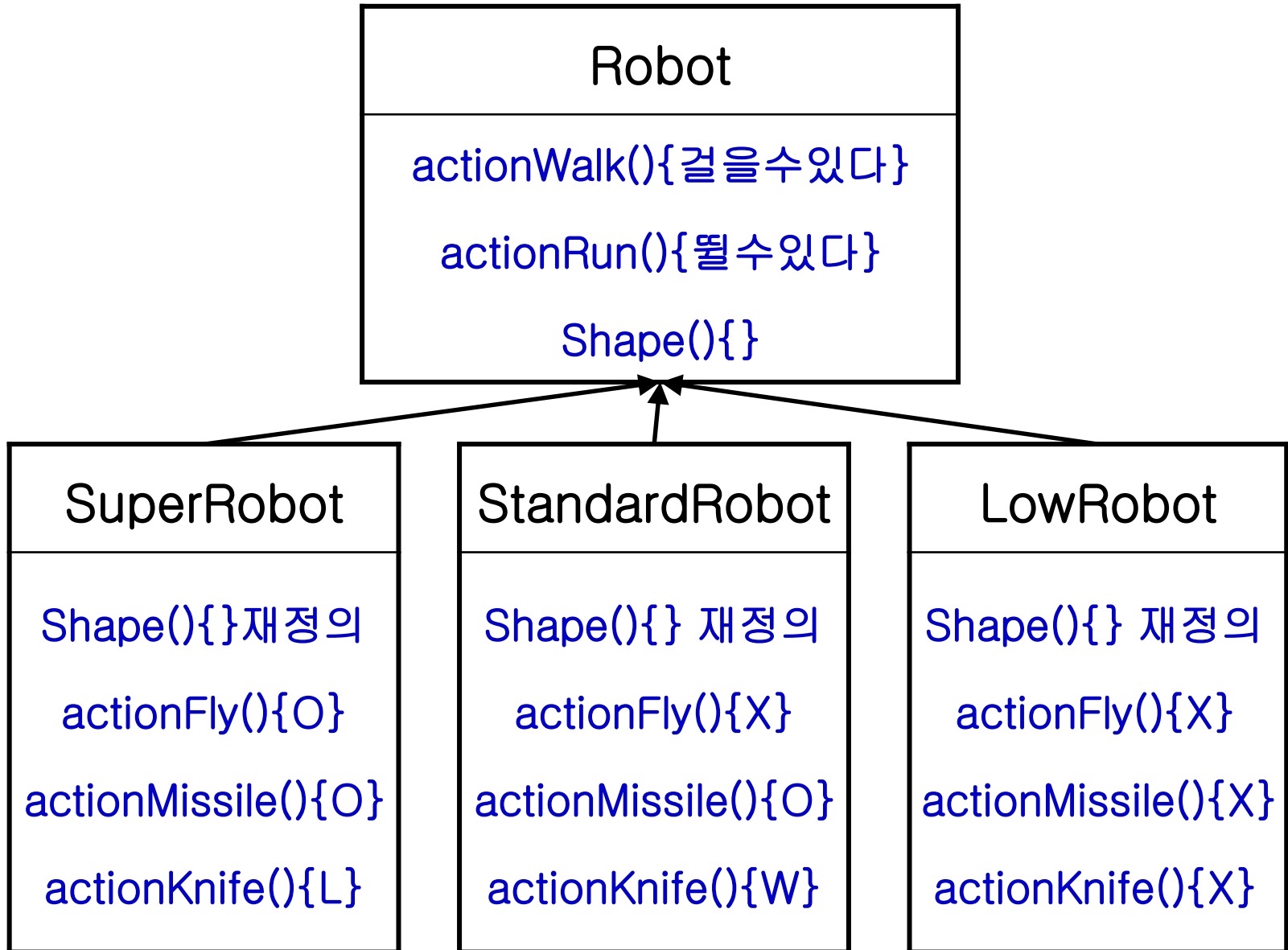
StandardRobot

```
Shape(){}  
actionWalk(){  
    걸을수있다}  
actionRun(){  
    뛸수있다}  
actionFly(){X}  
actionMissile(){O}  
actionKnife(){W}
```

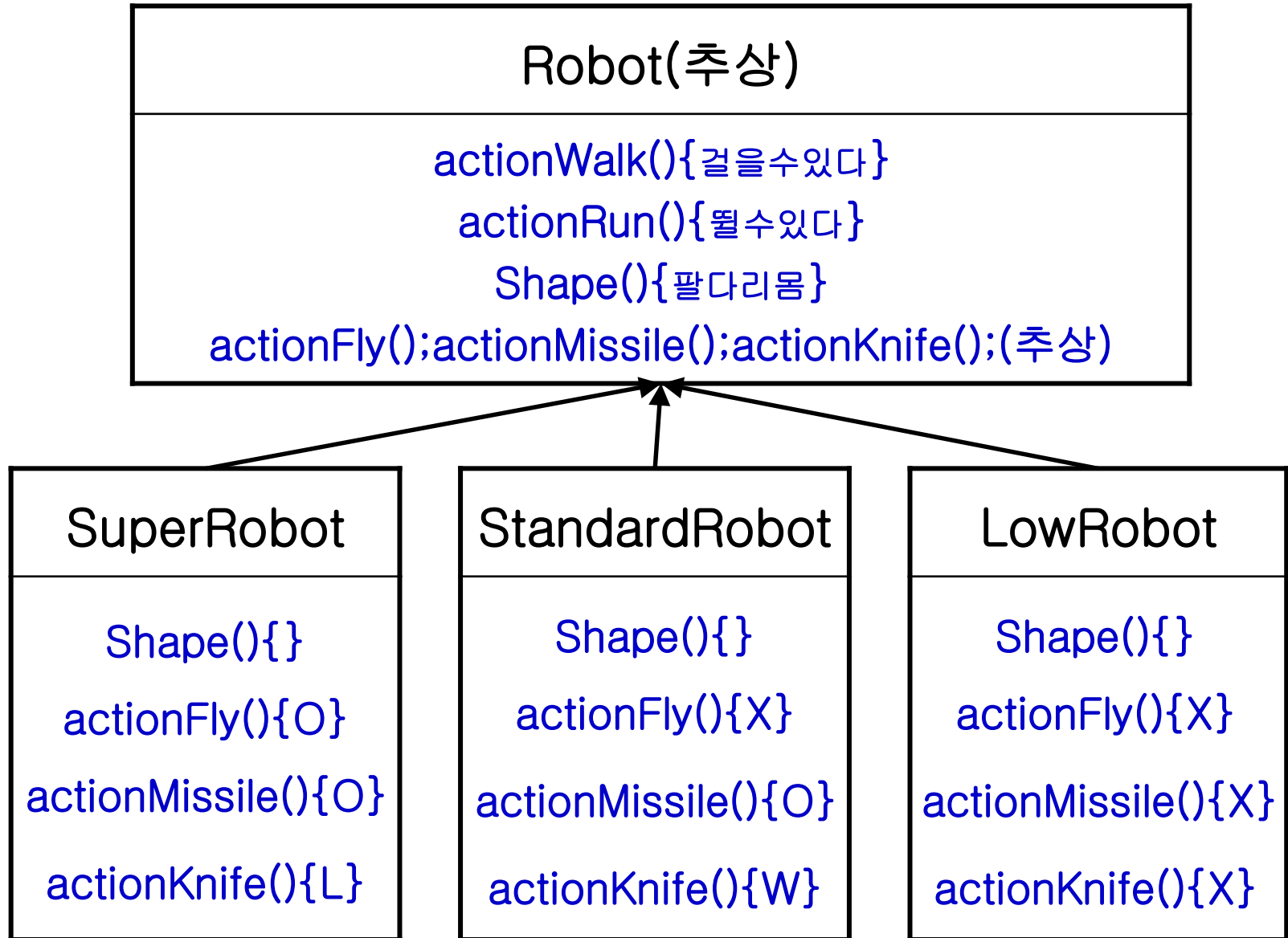
LowRobot

```
Shape(){}  
actionWalk(){  
    걸을수있다}  
actionRun(){  
    뛸수있다}  
actionFly(){X}  
actionMissile(){X}  
actionKnife(){X}
```

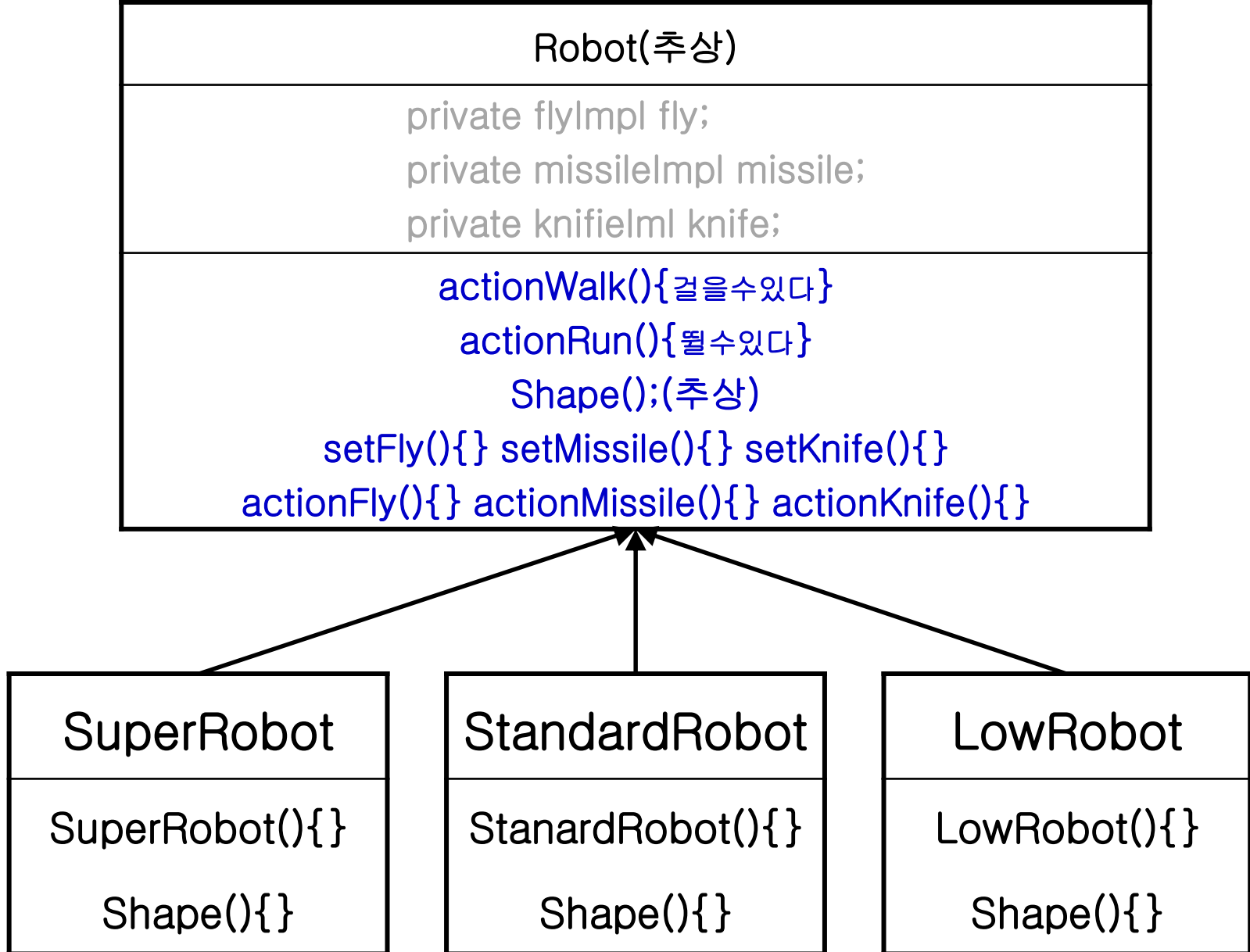
2단계. 공통점을 슈퍼클래스로



3단계. 공통점을 추상클래스로



5단계. 각기능을 객체(부품)화 ; object modularization



인터페이스 관련

