

2/14 study 😊

생성자 • New 연산자로 클래스로부터 객체를 생성할때 호출
되어 객체의 초기화를 담당.

```
CookieRunMain.java
1 package com.cookie.n1;
2
3 public class CookieRunMain {
4
5     public static void main(String[] args) {
6
7         GoldfishBread goldfish = new GoldfishBread();
8         |
9         System.out.println(goldfish.base);
10        System.out.println(goldfish.inside);
11    }
12 }
13 }
```

```
package com.cookie.n1;

public class GoldfishBread {

    String base;
    String inside;

    public GoldfishBread() {
        this.base="기본";
        this.inside="팥";
        System.out.println(this.base);
        System.out.println(this.inside);
    }
}
```

기본
팥



객체



기본 붕어빵 = 팥



바꿀 수 있음 > 슈크림/고구마..

생성자 오버로딩

- 매개변수만 바뀌어서 생성자를 여러 개 생성하는 것

```
public GoldfishBread() {  
    this.base="기본";  
    this.inside="달";  
    System.out.println(this.base);  
    System.out.println(this.inside);  
}  
  
public GoldfishBread(String base, String inside) {  
    System.out.println(base);  
    System.out.println(inside);  
}
```



```
public class CookieRunMain {  
    public static void main(String[] args) {  
        GoldfishBread goldfish = new GoldfishBread();  
        //System.out.println(goldfish.base);  
        //System.out.println(goldfish.inside);  
        GoldfishBread gfb2= new GoldfishBread("녹차", "고구마슈크림");  
    }  
}
```

접근지정자

	동일 클래스	동일한 패키지의 클래스	다른 패키지의 하위 클래스	다른 패키지의 클래스
Private	V			
(default)	V	V		
Protected	V	V	V	
public	V	V	V	V

public : 어디서나 아무나 접근 가능

protected: 상속관계라면 어디서나 접근 가능

default: 같은 패키지내에서만 아무나 접근 가능

private: 하나의 class(객체)내부에서 아무나 접근 가능, 외부에서 접근 불가, 자기 자신만 호출 가능.

Final / abstract

- void: return 값 X
- Final: 변경 불가의 값, 메서드앞에 오게 될 경우에는 오버라이딩이 불가, 클래스 앞에 오면 상속 불가
- Abstract: 추상화 / 변수앞에 올 수 없다. 선언부만 있고 구현부가 없음/ 서브 클래스에서 재정의 하기 위해 존재.
- Static: 같은 클래스들이 공유하는 공유변수 하나의 커다란 메모리 공간에 이어져있음

추상화

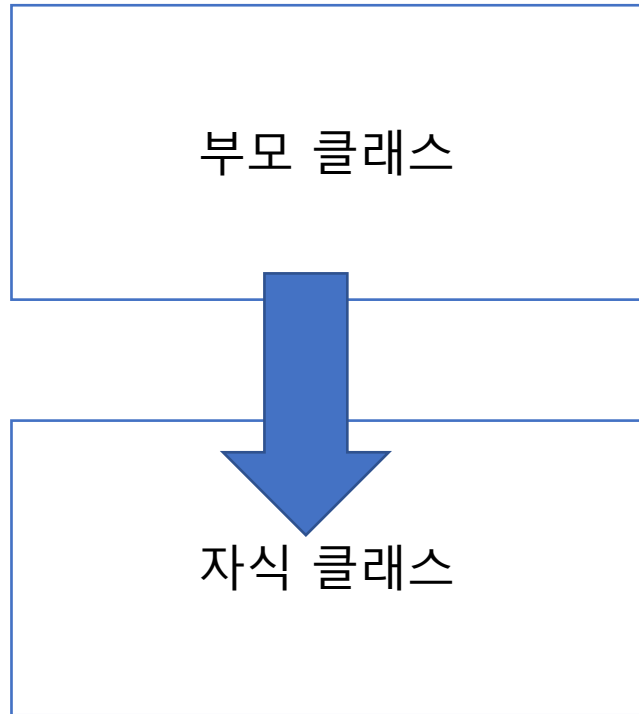
빠대로만 사용할 수 있게 클래스를 틀로 만드는 것

>abstract = new로 부를 수 없어요.

>abstract 선언의 순간 변수로 사용 불가에옴.

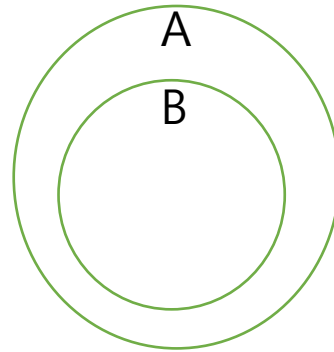
>추상 클래스 = 공통 자료형 공통으로 사용가능한 것을 만들어주는 거
구현시 새로운 class 생성> extends 상속 > 메인에서 new 로 호출...

상속, 다형성



Extends

- 별도의 선언 없이 부모클래스에 있는 변수들을 자유롭게
- 사용 가능하다.



다형성

1) 반드시 상속관계일 때 성립

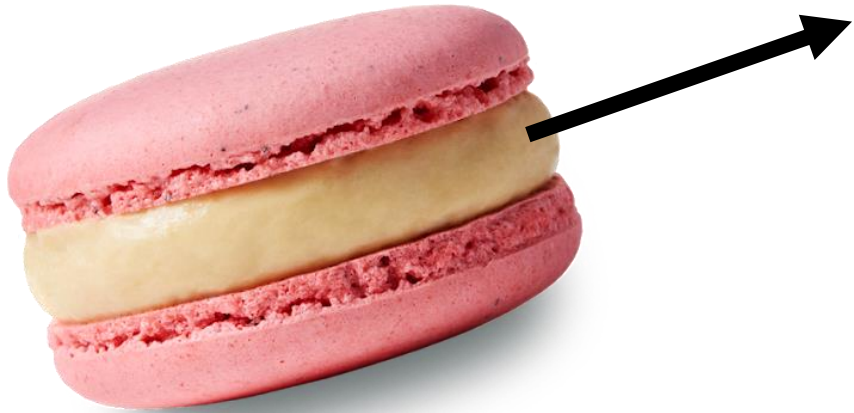
A is a B - extends, implement

A has a B - 멤버변수, 지역변수

반드시 상속이 있어야 사용가능.

주의: 부모타입 변수로는 자식 부분 접근 불가

인터페이스



인터페이스

- 코드를 분리하는 것
- 배터리분리/ 교체형 부품/ 프로그램의 부품을 만드는 것, 교체해서 바꾸어 착용

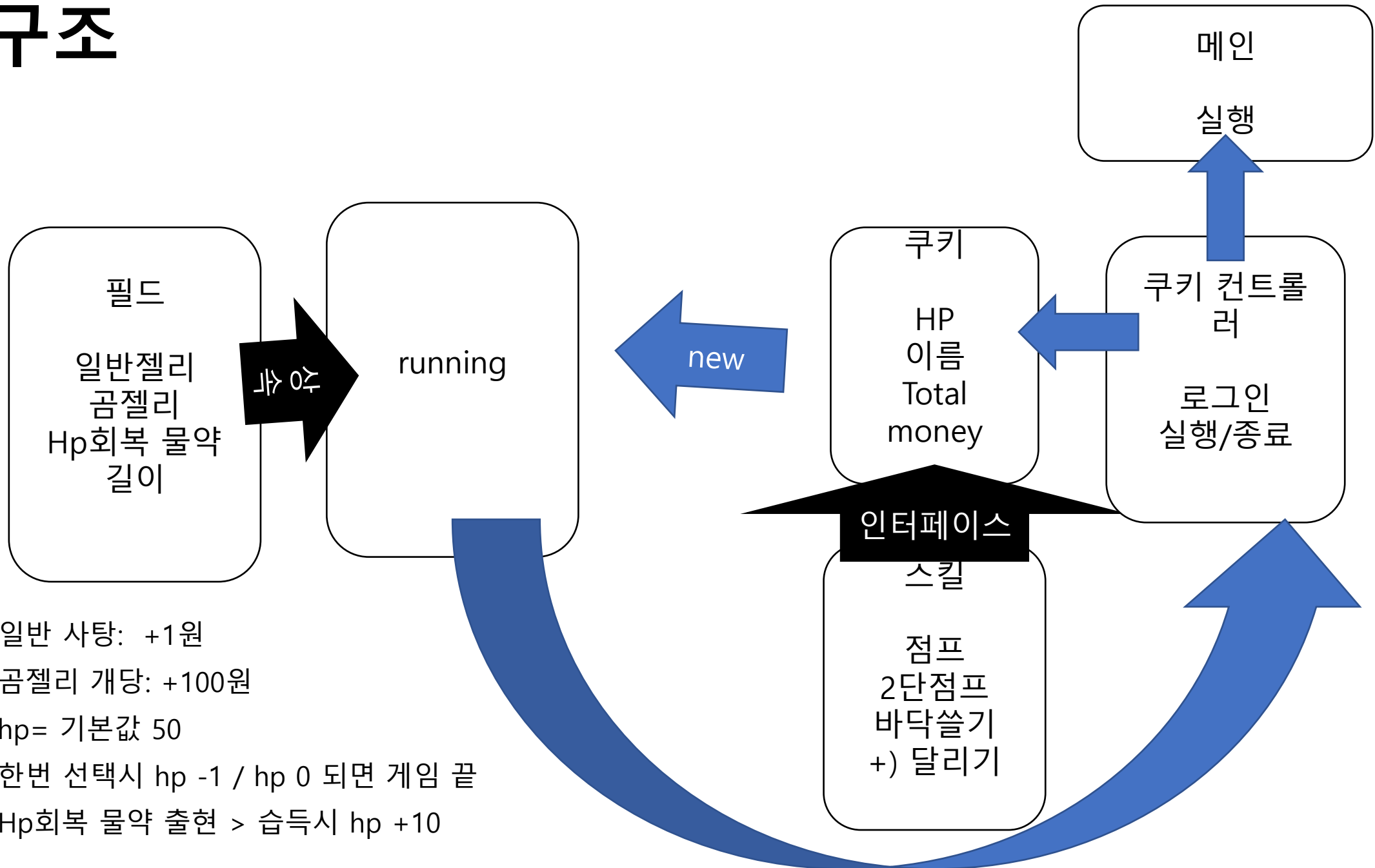
분리된 도구

> 인터페이스는 분리되는 부품과 같은 것. 약속된 인터페이스로 제품을 만들어요

인터페이스는 구현한 객체를 꽂아넣는것, 부품장착 캡슐화

- 모든 변수는 public/static/final
- 모든 메서드는 public abstract(기능이X)

구조



- 일반 사탕: +1원
- 곰젤리 개당: +100원
- hp= 기본값 50
- 한번 선택시 hp -1 / hp 0 되면 게임 끝
- Hp회복 물약 출현 > 습득시 hp +10