

Day12

Day11

Day11

- 메서드 오버라이딩
 - 가상메서드
 - 어노테이션
- 다형성
- 업/다운캐스팅
- instanceof
- 추상클래스 abstract
- 인터페이스 interface

메소드 오버로딩

- 메소드 오버로딩(Overloading)
 - ▣ 이름이 같은 메소드 작성
 - 매개변수의 개수나 타입이 서로 다르고
 - 이름이 동일한 메소드들
 - ▣ 리턴 타입은 오버로딩과 관련 없음

// 메소드 오버로딩이 성공한 사례

```
class MethodOverloading {  
    public int getSum(int i, int j) {  
        return i + j;  
    }  
    public int getSum(int i, int j, int k) {  
        return i + j + k;  
    }  
}
```

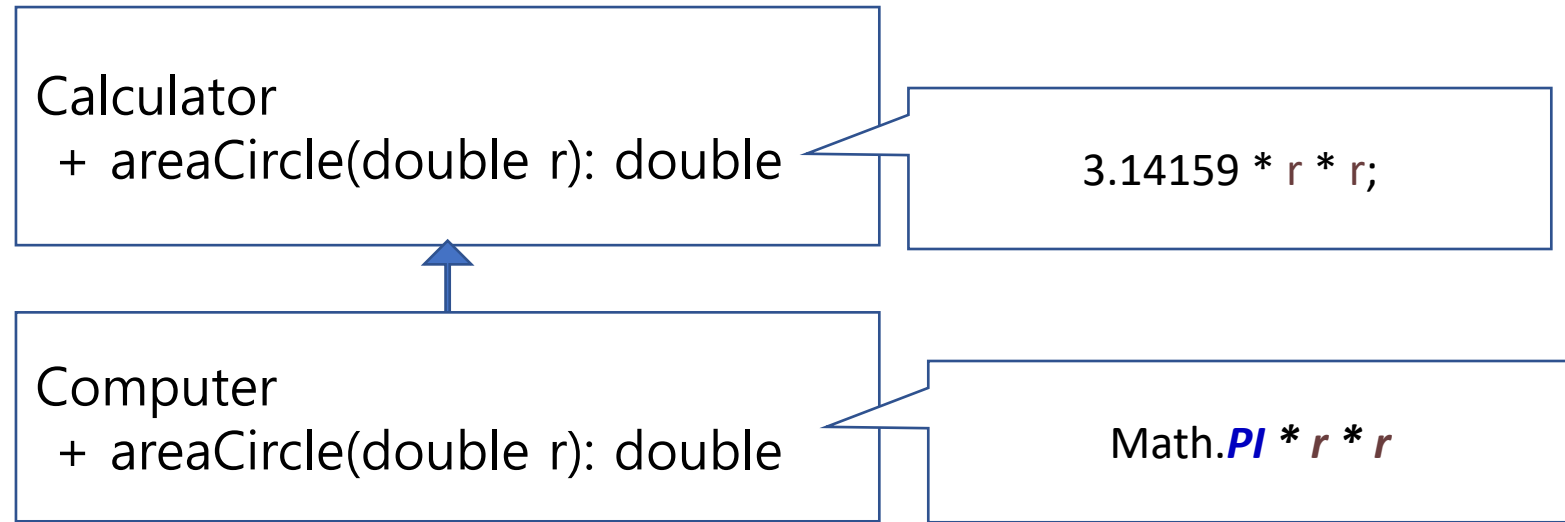
// 메소드 오버로딩이 실패한 사례

```
class MethodOverloadingFail {  
    public int getSum(int i, int j) {  
        return i + j;  
    }  
    public double getSum(int i, int j) {  
        return (double)(i + j);  
    }  
}
```

두 개의 getSum() 메소드는 매
개변수의 개수, 타입이 모두 같
기 때문에 메소드 오버로딩 실패

실습: 메서드 오버라이딩

- Calculator 를 상속받는 Computer



Main:

```
int r = 10;
```

```
Calculator calculator = new Calculator();
```

```
System.out.println(" 원면적: " + calculator.areaCircle(r));
```

```
System.out.println();
```

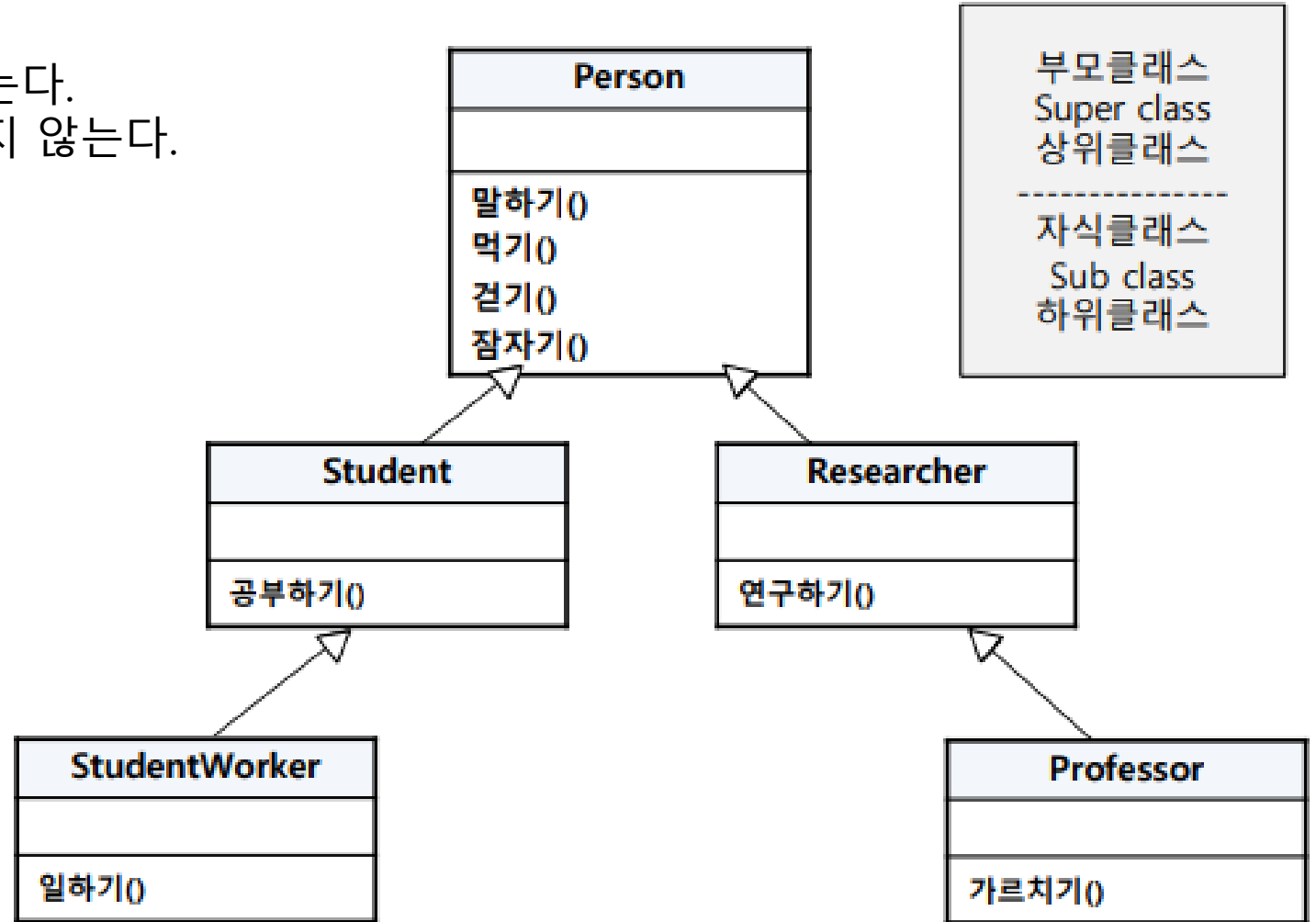
```
Computer computer = new Computer();
```

```
System.out.println(" 원면적: " + computer.areaCircle(r));
```

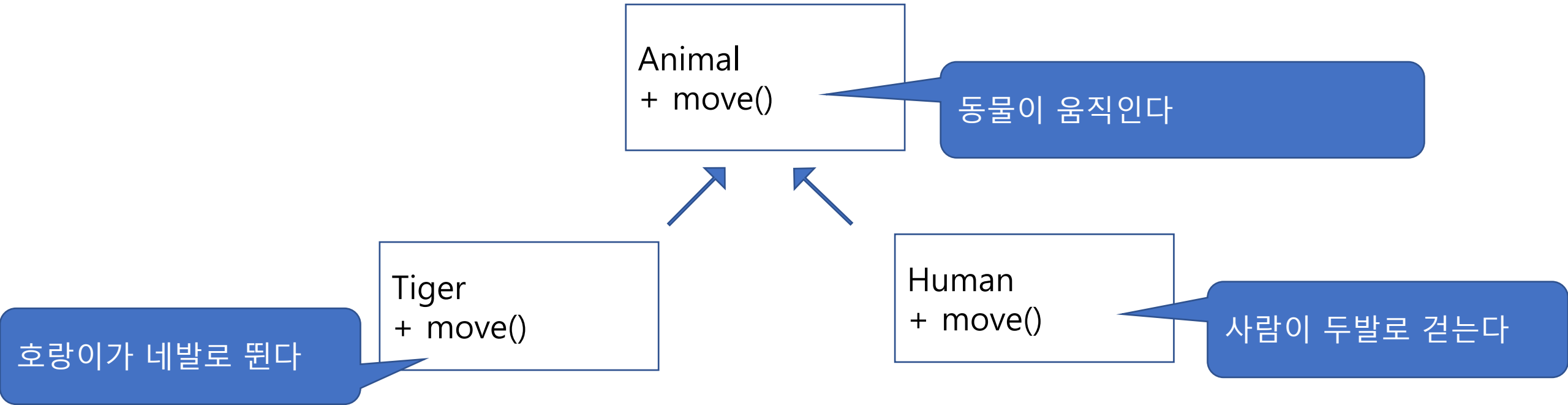
다형성

자바상속의 특징

- 자바에서는 다중 상속을 지원하지 않는다.
- 자바에서는 상속의 횟수에 제한을 두지 않는다.
- 자바에서 계층구조의 최상위에 있는 클래스는 java.lang.Object 이다.

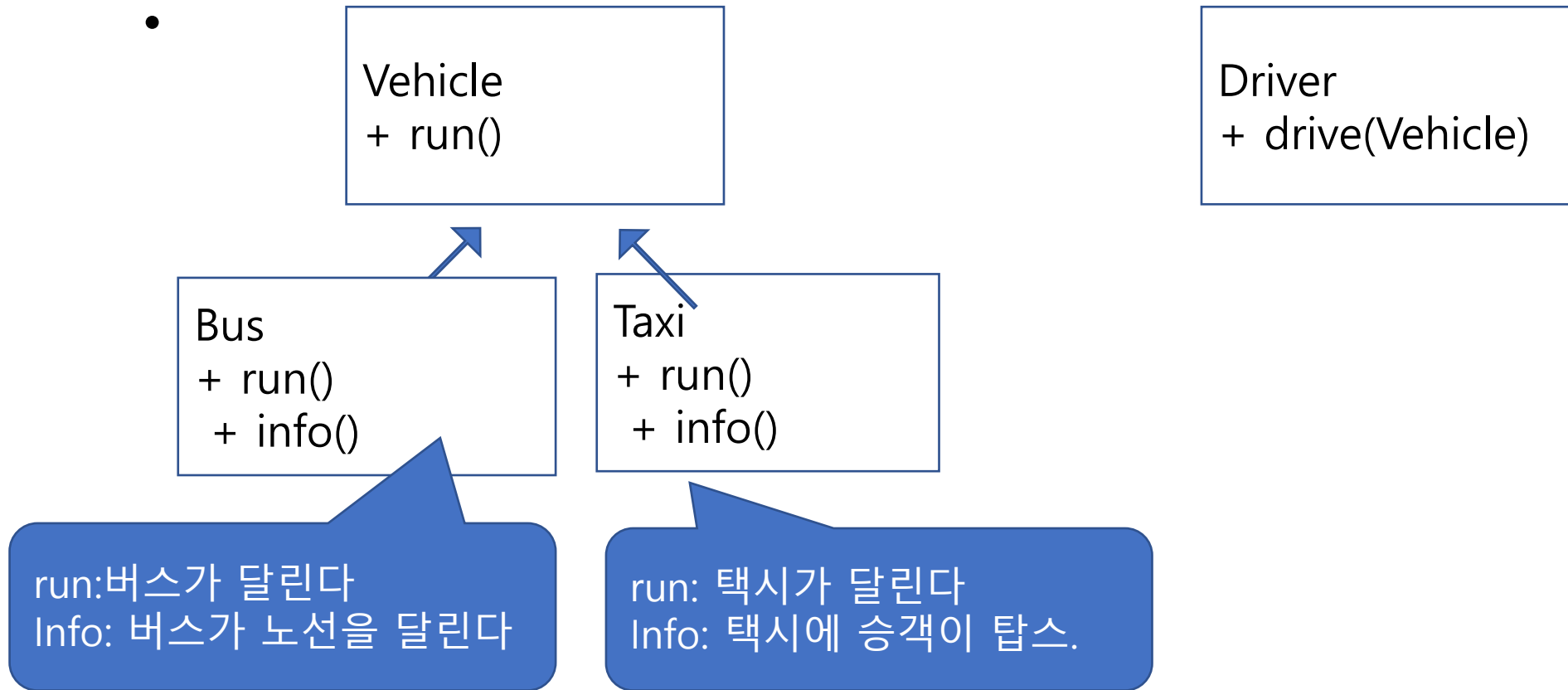


실습: 다형성



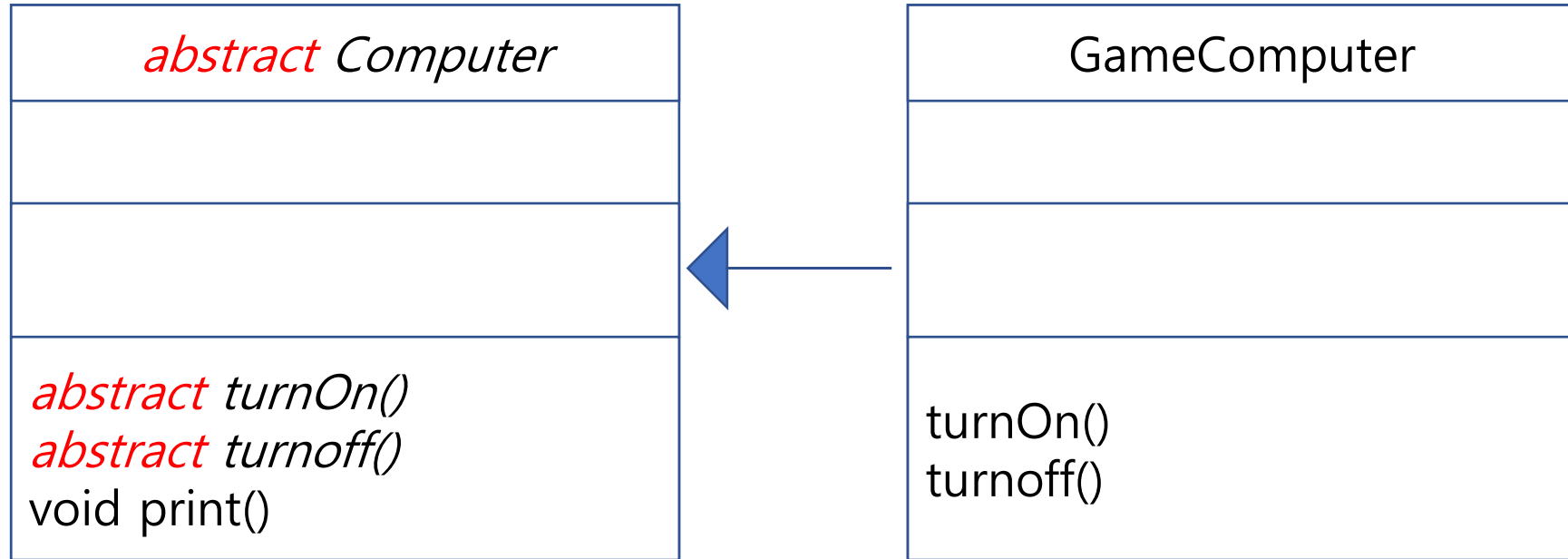
실습: 다형

- Main class 에서 ArrayList 를 사용해서 5대의 Bus, Taxi에서 info() 를 사용해 보자



실습: 추상 클래스/메서드

- P280



실습: 추상 클래스/메서드

- P280

```
public abstract class Computer {  
  
    abstract void display();  
    abstract void typing();  
  
    public void print() {  
        System.out.println("");  
    }  
  
}
```

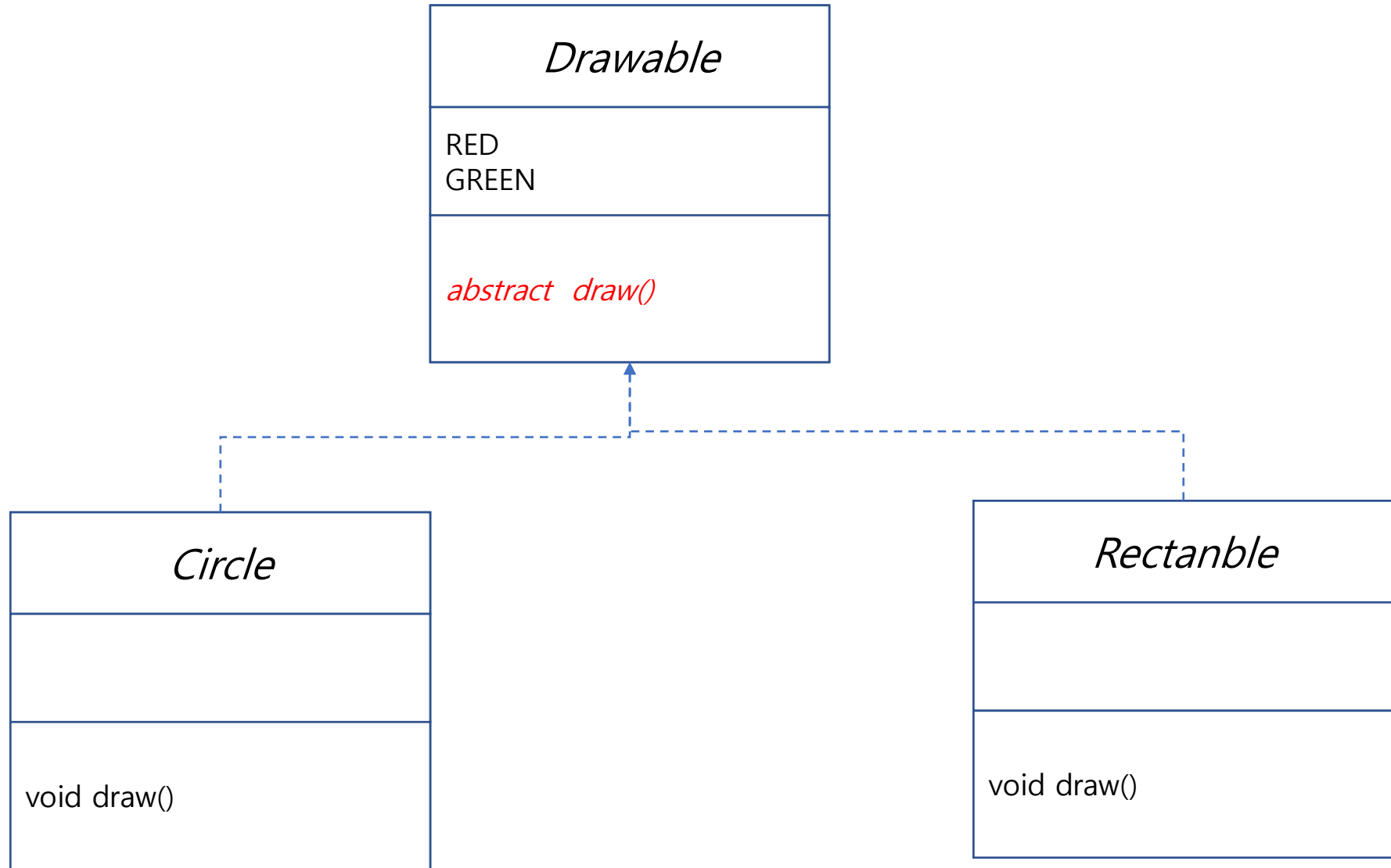
```
public class ComputerMain {  
  
    public static void main(String[] args) {  
  
        Computer c1 = new Computer();  
        Computer c2 = new  
        c2.display();  
        c2.typing();  
        //  
        c2.print();  
    }  
  
    public class ComputerMain {  
  
        public static void main(String[] args) {  
  
            // Computer c1 = new Computer();  
            Computer c2 = new Desktop();  
            c2.display();  
            c2.typing();  
            //  
            c2.print();  
        }  
  
    }  
  
}
```

Cannot instantiate the type Computer

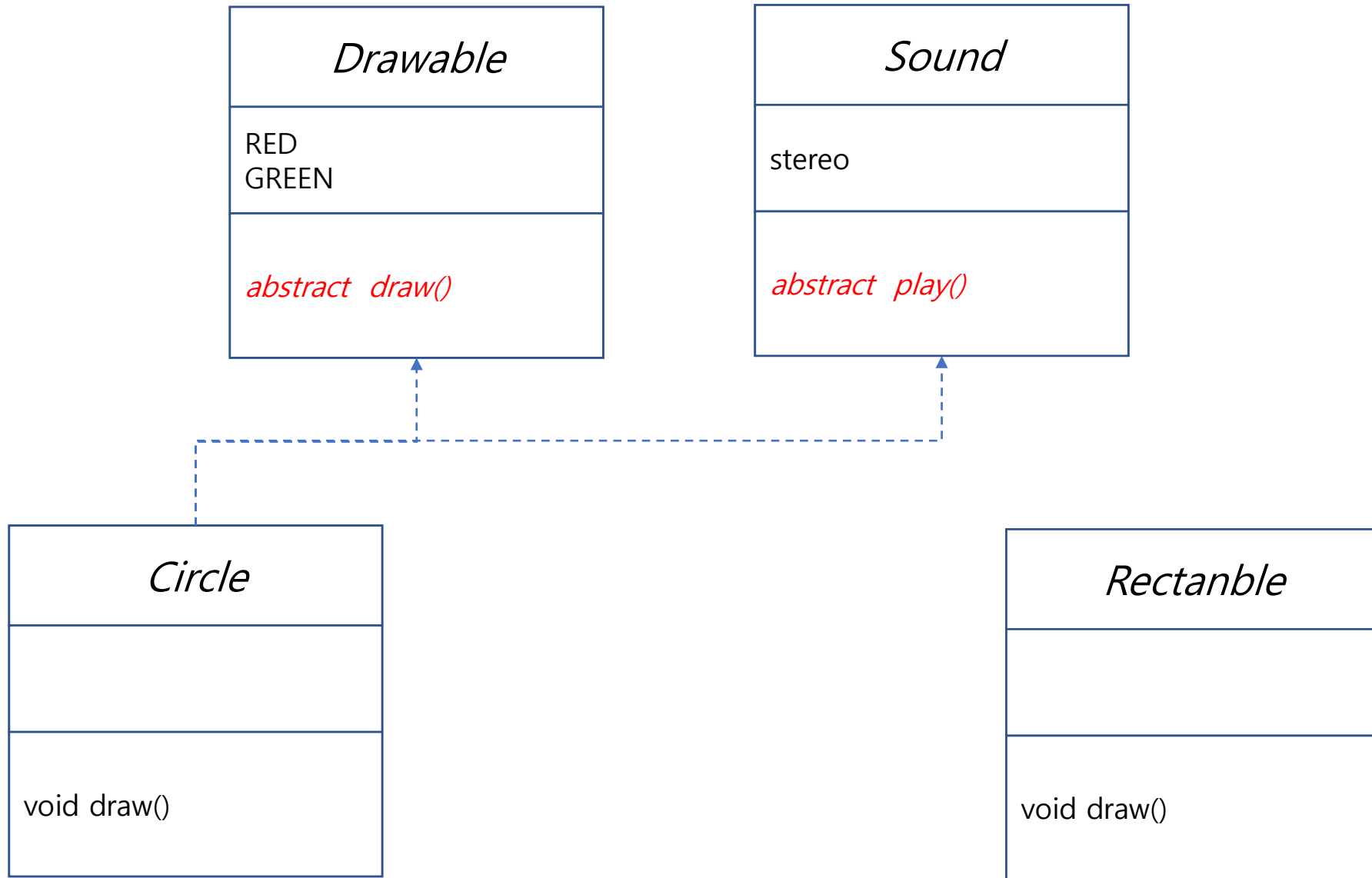
실습:

- 추상클래스 실습 Kr.chapter9.ex
- 인터페이스 실습 Kr.chapter10.ex

실습: 인터페이스



실습: 인터페이스



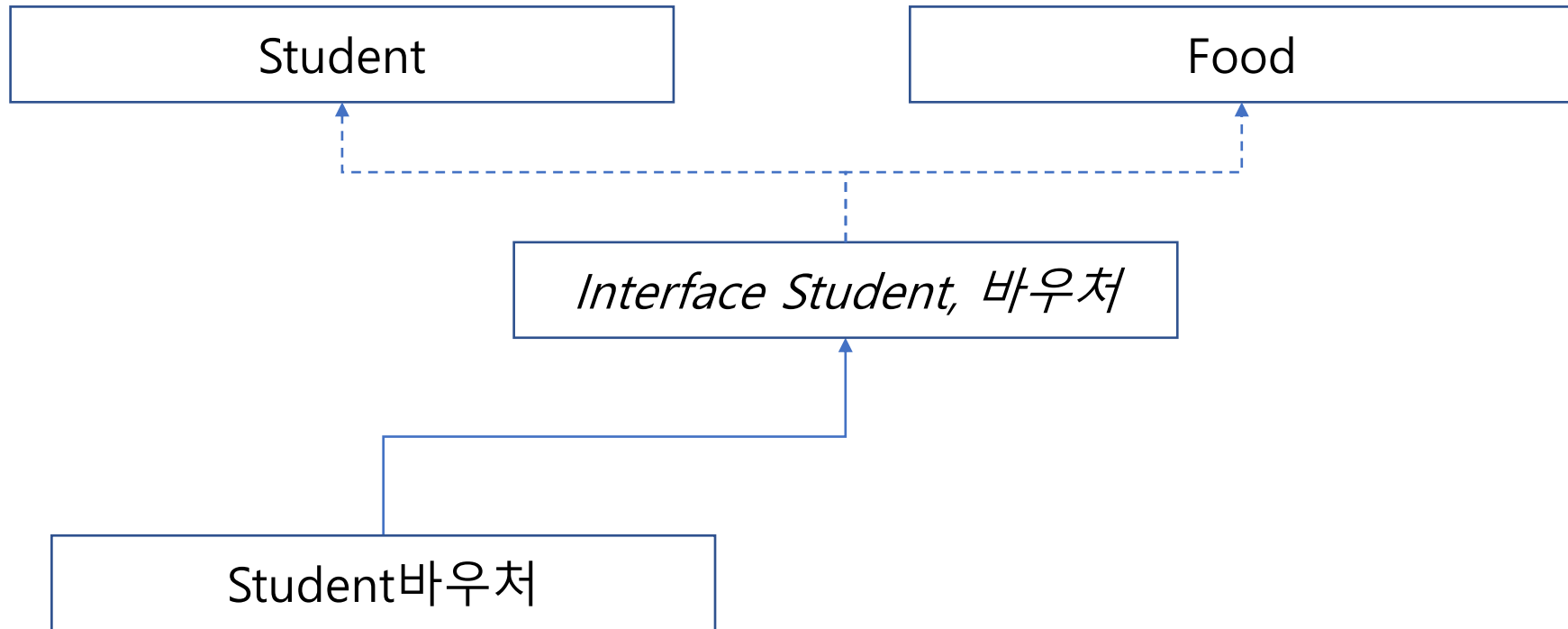
인터페이스

- P314



인터페이스

- P314



Day 12, 5/28

5/28일 Day12

- Ch11 P351 기본 클래스
 - 컬렉션
 - 내부 클래스
- Ch7 P199 배열/객체복사
- 실습?

Object class

- <https://docs.oracle.com/javase/10/docs/api/java/lang/Object.html>
- toString() 재정의

Object class

- <https://docs.oracle.com/javase/10/docs/api/java/lang/Object.html>

: equals()

- ▶ 두 객체의 비교시 == 와 Object 클래스의 equals() 메서드를 사용한다
- ▶ == 와 equals()의 차이 : **확실히 구분**하여 사용
 - ▶ == 참조변수값 비교
 - ▶ equals() : 정의한 값 비교
- ▶ 참조 변수값을 먼저 비교한다
- ▶ 참조변수값이 같으면 두 객체는 같은 것으로 한다
- ▶ 참조변수값이 다르면 두 객체의 속성값을 비교한다

Object class

- <https://docs.oracle.com/javase/10/docs/api/java/lang/Object.html>
- == 와 equals(), hashCode()

```
@Override
public boolean equals(Object obj) {
    if (obj instanceof Student) {
        Student std = (Student) obj;
        if (studentId == std.studentId)
            return true;
        else
            return false;
    }
    return false;
}

@Override
public int hashCode() {
    return x;
}
```

Object class

- <https://docs.oracle.com/javase/10/docs/api/java/lang/Object.html>
- clone(), 얕은복사/깊은복사 P213~P217

- ▶ 객체 복제 : 원본 객체의 값과 동일한 값을 가진 새로운 객체를 생성하는 것
- ▶ 얕은 복제 : 단순히 필드 값을 복사하는 방식으로 객체를 복제하는 것
- ▶ Cloneable 인터페이스를 구현하여 clone 메서드를 사용 가능하게 해야 한다

```
public class Point implements Cloneable {  
    private int x;  
    private int y;  
  
    // ...  
    public Point getClone() {  
        Point clone = null;  
        try {  
            clone = (Point)clone();  
        } catch (CloneNotSupportedException e) {}  
        return clone;  
    }  
}
```

System class

- <https://docs.oracle.com/javase/10/docs/api/java/lang/System.html>

System 클래스 용도

■ 운영체제의 기능 일부 이용 가능

- 프로그램 종료, 키보드로부터 입력, 모니터 출력, 메모리 정리, 현재 시간 읽기
- 시스템 프로퍼티 읽기, 환경 변수 읽기

`static void`

`gc()`

Runs the garbage collector.

`static Map<String,String> getenv()`

Returns an unmodifiable string map view of the current system environment.

`static String`

`getenv(String name)`

Gets the value of the specified environment variable.

`static String`

`getProperty
(String key,
String def)`

Gets the system property indicated by the specified key.

Garbage Collection

Garbage Collection

- 객체 소멸을 다루는 JVM 메커니즘

객체의 소멸과 가비지 컬렉션

- 객체 소멸
 - ▣ new에 의해 할당된 객체 메모리를 자바 가상 기계의 가용 메모리로 되돌려 주는 행위
- 자바 응용프로그램에서 임의로 객체 소멸할 수 없음
 - ▣ 객체 소멸은 자바 가상 기계의 고유한 역할
 - ▣ 자바 개발자에게는 매우 다행스러운 기능
 - C/C++에서는 할당받은 객체를 개발자가 되돌려 주어야 함
 - C/C++ 프로그램 작성을 어렵게 만드는 요인
- 가비지
 - ▣ 가리키는 레퍼런스가 하나도 없는 객체
 - 누구도 사용할 수 없게 된 메모리
- 가비지 컬렉션
 - ▣ 자바 가상 기계의 가비지 컬렉터가 자동으로 가비지 수집 반환

Garbage Collection 정리

- 가비지 컬렉션
 - ▣ 자바에서 가비지를 자동 회수하는 과정
 - 가용 메모리로 반환
 - ▣ 가비지 컬렉션 스레드에 의해 수행
- 개발자에 의한 강제 가비지 컬렉션
 - ▣ System 또는 Runtime 객체의 gc() 메소드 호출

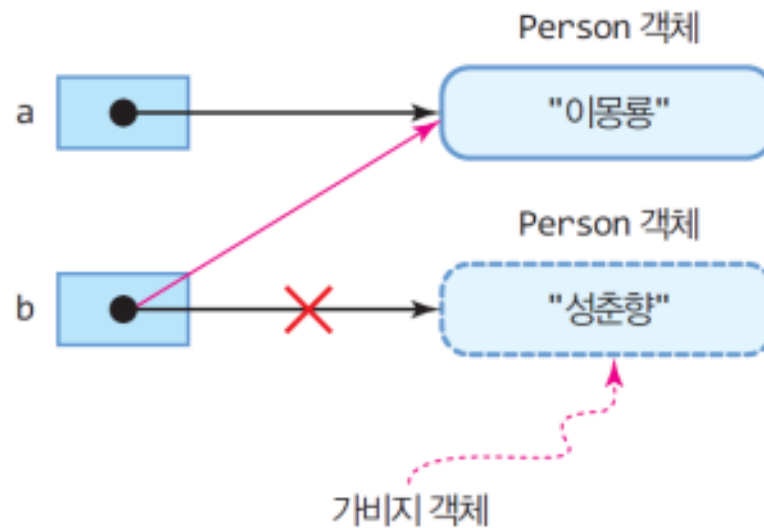
```
System.gc(); // 가비지 컬렉션 작동 요청
```

- 이 코드는 자바 가상 기계에 강력한 가비지 컬렉션 요청
 - 그러나 자바 가상 기계가 가비지 컬렉션 시점을 전적으로 판단

Garbage Collection

- 소멸 대상

```
Person a, b;  
a = new Person("이몽룡");  
b = new Person("성춘향");  
b = a; // b가 가리키던 객체는 가비지가 됨
```

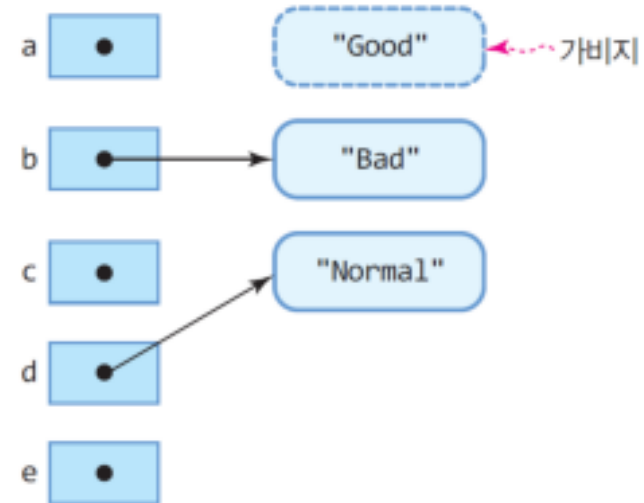


실습: Garbage Collection

```
public class GarbageEx {  
    public static void main(String[] args) {  
        String a = new String("Good");  
        String b = new String("Bad");  
        String c = new String("Normal");  
        String d, e;  
        a = null;  
        d = c;  
        c = null;  
    }  
}
```



(a) 초기 객체 생성 시(라인 6까지)



(b) 코드 전체 실행 후

실습: Garbage Collection

```
class Employee {
    int eno;

    public Employee(int eno) {
        this.eno = eno;
        System.out.println("Employee" + eno + " 생성");
    }

    @Override
    protected void finalize() throws Throwable {
        System.out.println("Employee" + eno + " 삭제");
    }
}
```

```
public class SystemGcExample {
    public static void main(String[] args) {
        Employee emp = new Employee(1);
        emp = null;
        emp = new Employee(2);
        emp = new Employee(3);
        System.out.println("Employee " + emp.eno );
        System.gc();
    }
}
```

System class

- getProperty()

키 (Key)	설명	값 (value)
java.version	자바의 버전	1.8.0_20
java.home	사용하는 JRE의 파일 경로	<jdk 설치경로>\jre
os.name	Operating System name	Window 10
file.separator	File separator("\\" on window)	\
user.name	사용자의 이름	사용자 계정
user.home	사용자의 홈 디렉토리	C:\Users\사용자 계정
user.dir	사용자가 현재 작업 중인 디렉토리 경로	다양

```
// getProperty()  
String osName = System.getProperty("os.name");  
String userName = System.getProperty("user.name");
```

Format string

- <https://interconnection.tistory.com/116>

홍길동 100 (%), 수학 100, 과학 300

..

..

Day13 - 5/31 진행

- 제네릭
 - 교재 P388-403
 - 생활코딩
 - <https://opentutorials.org/course/1223/6237>
- 컬렉션 프레임워크
- 예외처리