안드로이드 스터디 부록: Java and C++

2018-2

유어슈 개발팀 조성재

이번 PPT에서는

차이점은 많지만... 당장 안드로이드 개발 시 유의점 2개 정도 담았습니다.

C++ 알고 계실 테니, 코드에 있는 내용은 (조금이라도) 이해하실 줄로 믿습니다!

Java는 객체를 좋아해요 (1)

- Java는 OOP의 개념을 강제하는 편임.
- C++의 경우 Class 외부에 데이터, 함수를 정의할 수 있음.
- Java의 경우 Class 외부에 데이터, 함수를 정의할 수 없음. 반드시 Class 내부에 존재해야 함; 프로그램을 객체들의 상호작용으로 볼 수 있음.
  - main함수는 정의된 Class들 중 하나에서 멤버함수로 정의됨.
  - 함수를 인자로 넘겨줄 때에는 Class로 감싸서 넘겨주는 방식을 사용함. (이 특징은 안드로이드에서 확인!)

Java는 객체를 좋아해요 (2)

```java
package com.company;

// Violations:
// int data;
// System.out.println("안녕하세요!");
/* void main(String[] args) {

} */

public class Main {
    private int data;
    public static void main(String[] args) {
        // write your code here
        System.out.println("안녕하세요!");
        main2();
    }

    private static void main2(String[] args) {
        System.out.println("YEE!");
    }
}
```

```cpp
#include <iostream>

using namespace std;

// Who cares?
int data;
void i_am_main(int argc, char **argv) {
    std::cout << "YEE!" << std::endl;
}

int main(int argc, char **argv) {
    std::cout << "Hello, World!" << std::endl;

    void (*func)(int, char**);
    func = i_am_main;

    func(argc, argv);

    return 0;
}
```

Java의 Reference 형 (1)

- Java의 int, double, char과 같은 primitive 자료형을 제외한 자료형은 모두 참조(Reference)형임.
- 참조형 변수는 참조형 객체를 가리킴.
  - 일종의 포인터 역할.
  - Java에서 “MyClass m”은 Reference형 변수를 선언한 것임; C++과 같이 Stack에 생기지 않음 → new를 사용하여 새 객체를 만들어 가리키거나, 이미 가리키고 있는 참조형 변수의 값을 복사하여 가리킬 수 있음.
  - new는 있지만 free/delete는 없음.
  - 함수 파라미터로 받은 Reference형 변수: 함수 내부에서 수정하면 그 객체도 수정됨.

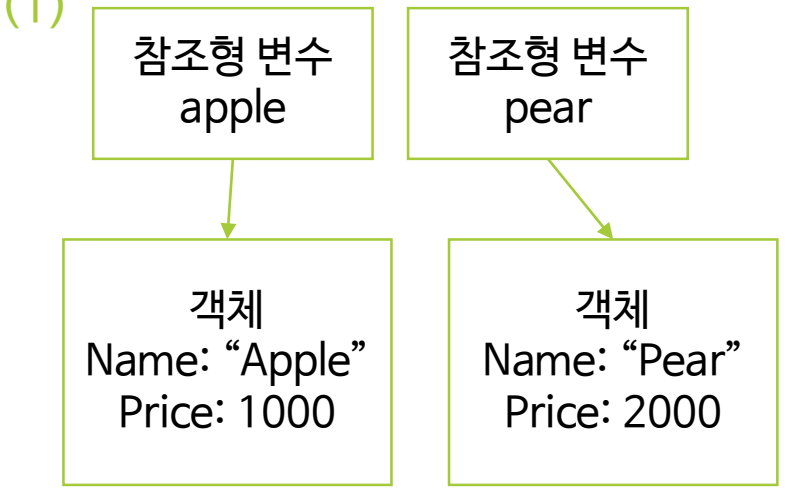
Java의 Reference 형 (2)

```java
package com.company;

class Fruit {
    private String name;
    private int price;

    public Fruit(String name, int price) {
        this.name = name;
        this.price = price;
    }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public void printFruitInfo() {
        System.out.println("Fruit " + name);
        // string concatenation!
        System.out.println("price: " + price);
    }
}
```

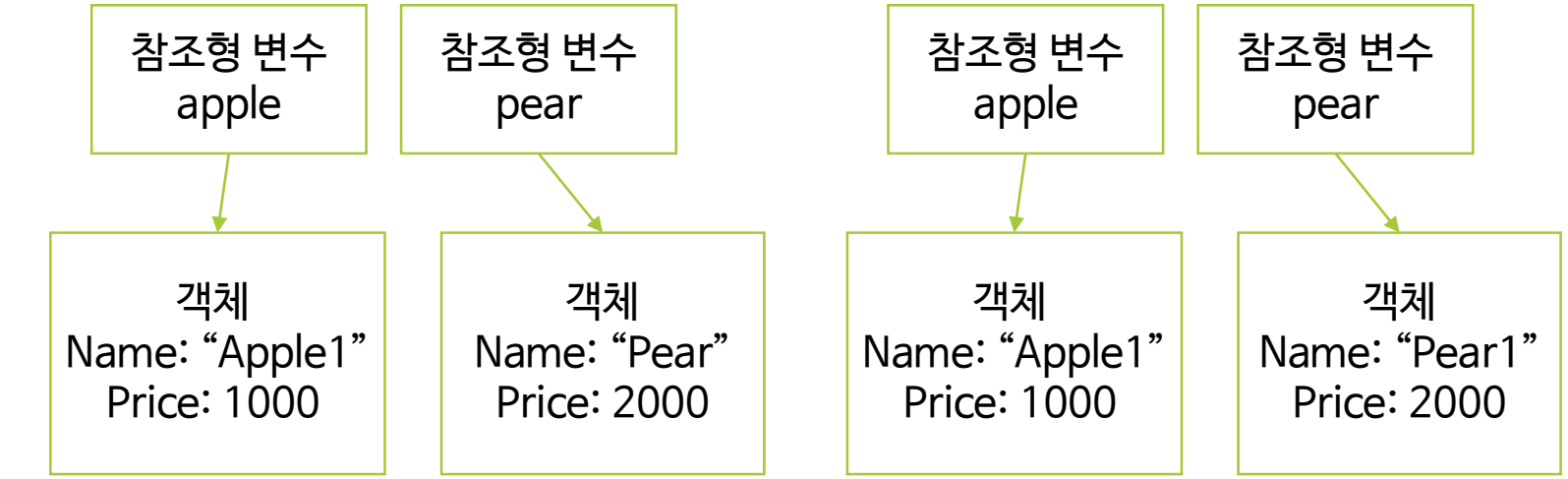
```java
public class Main {

    public static void main(String[] args) {
    // write your code here
        Fruit apple = new Fruit("Apple", 1000);
        Fruit pear = new Fruit("Pear", 2000);

        changeFruit(apple);
        changeFruit(pear);

        apple.printFruitInfo();
        pear.printFruitInfo();

        pear = new Fruit("Pear1", 2000);
        pear.printFruitInfo();
    }

    private static void changeFruit(Fruit fruit) {
        if (fruit.getName().equals("Apple")) {
            fruit.setName("Apple1");
        }
    }
}
```

```java
public class Main {

    public static void main(String[] args) {
    // write your code here
(1)     Fruit apple = new Fruit("Apple", 1000);
        Fruit pear = new Fruit("Pear", 2000);

(2)     changeFruit(apple);
        changeFruit(pear);

        apple.printFruitInfo();
        pear.printFruitInfo();

(3)     pear = new Fruit("Pear1", 2000);
        pear.printFruitInfo();
    }

    private static void changeFruit(Fruit fruit) {
        if (fruit.getName().equals("Apple")) {
            fruit.setName("Apple1");
        }
    }
}
```

(1)



(2)



(3) JVM에 의해 메모리 수거예정