

```

package java2.interfaceEx.cook;

import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {

        //Cook형을 담을 수 있는 ArrayList를 준비한다
        ArrayList<Cook> list = new ArrayList<Cook>();

        // Cook을 상속받았으므로 각 각의 요리사 클래스를 담을 수 있다
        list.add(new Dy("곽단야"));
        list.add(new Mg("김민규"));
        list.add(new Mj("김민지"));
        list.add(new Bj("김병진"));
        list.add(new Hs("김현수"));
        list.add(new Nj("문나정"));
        list.add(new Tm("박태민"));
        list.add(new Yjin("서예진"));
        list.add(new Ly("이윤"));
        list.add(new LYJ("이윤정"));
        list.add(new LJH("이정훈"));
        list.add(new EK("조은경"));
        list.add(new Juntae("표준태"));
        list.add(new JH("허재혁"));

        // 요리사별로 ArrayList 준비하기
        ArrayList<한식가능한> 한식요리사 = new ArrayList<한식가능한>();
        ArrayList<중식가능한> 중식요리사 = new ArrayList<중식가능한>();
        ArrayList<양식가능한> 양식요리사 = new ArrayList<양식가능한>();

        // 요리사종류별로 목록을 구성한다.
        for (Cook cook : list) {
            if (cook instanceof 한식가능한) {
                한식요리사.add((한식가능한) cook);
            } else if (cook instanceof 중식가능한) {
                중식요리사.add((중식가능한) cook);
            } else if (cook instanceof 양식가능한) {
                양식요리사.add((양식가능한) cook);
            }
        }

        System.out.println("—한식 요리사 입니다—");
        System.out.println(한식요리사);

        System.out.println("—중식 요리사 입니다—");
        System.out.println(중식요리사);

        System.out.println("—양식 요리사 입니다—");
        System.out.println(양식요리사);

        // 랜덤한 값 얻어오기
        int random = (int) (Math.random() * 3); // 0~2;
    }
}

```

```
//선택한 요리사종류 출력하기
String[] kinds = { "한식가능한", "중식가능한", "양식가능한" };
String selectedKind = kinds[random];
System.out.println("=====선택된 요리사 kind =====" + selectedKind);
```

```
//선택된 요리사종류 중에서 한 명의 요리사 선택하기
int randomRange = 0;
int finalOne;
Cook fianlCook = null;
```

```
switch (selectedKind) {
case "한식가능한":
    for (한식가능한 cook : 한식요리사) {
        System.out.println(cook);
    }
    randomRange = 한식요리사.size();
    System.out.println("randomRange="+ randomRange);
    finalOne = (int) (Math.random() * randomRange);
    fianlCook = (Cook) 한식요리사.get(finalOne);
    System.out.println("=====>최종선택");
    System.out.println(fianlCook);
    break;
```

```
case "중식가능한":
    for (중식가능한 cook : 중식요리사) {
        System.out.println(cook);
    }
    randomRange = 중식요리사.size();
    System.out.println("randomRange="+ randomRange);
    finalOne = (int) (Math.random() * randomRange);
    fianlCook = (Cook) 중식요리사.get(finalOne);
    System.out.println("=====>최종선택");
    System.out.println(fianlCook);
    break;
```

```
case "양식가능한":
    for (양식가능한 cook : 양식요리사) {
        System.out.println(cook);
    }
    randomRange = 양식요리사.size();
    System.out.println("randomRange="+ randomRange);
    finalOne = (int) (Math.random() * randomRange);
    fianlCook = (Cook) 양식요리사.get(finalOne);
    System.out.println("=====>최종선택");
    System.out.println(fianlCook);
    break;
```

```
}
```

```
}
```

```
}
```

```
class Cook{
    String name;
    public void 썰기(){
        System.out.println(" 잘 썰다");
    }
    public void 굽기(){
        System.out.println(" 잘 굽는다");
    }
    public void 튀기기(){
        System.out.println(" 잘 튀긴다");
    }
    public void 볶기(){
        System.out.println(" 잘 볶는다");
    }

    public Cook( String name){
        this.name = name;
    }

    @Override
    public String toString() {
        return "Cook [name=" + name + "]";
    }
    public Cook() {
        // TODO Auto-generated constructor stub
    }

}
```

//양식가능한 인터페이스를 구현한 요리사 HrCook 클래스

```
public class HrCook extends Cook implements 양식가능한{

    public HrCook(String name) {
        super(name);
        // TODO Auto-generated constructor stub
    }

    @Override
    public String 스파게티만들기() {
        // TODO Auto-generated method stub
        return "스파게티임";
    }

    @Override
    public String 피자만들기() {
        // TODO Auto-generated method stub
        return "피자임";
    }

    @Override
    public String 돈까스만들기() {
        // TODO Auto-generated method stub
        return "돈까스임";
    }

}
```

```
interface 양식가능한{
    public String 스파게티만들기();
    public String 피자만들기();
    public String 돈까스만들기();
}
```

```
interface 중식가능한{
    public String 마라탕만들기();
    public String 탕수육만들기();
    public String 짜장면만들기();
}
```

```
interface 한식가능한{
    public String 된장찌게만들기();
    public String 불고기만들기();
    public String 김밥만들기();
}
```