

```
interface 한식가능한{
    String 비빔밥만들기();
}
```



```
interface 중식가능한{
    String 탕수육만들기();
}
```



```
interface 일식만들기{
    String 초밥만들기();
}
```



```
Class Cook{
    String name;
}
```



오늘의 요리사 찾기

```
Class SampleCook extends Cook implements 일식가능한{
}
```



1. 각 기 다른 요리사클래스를 Cook배열에 저장하기

```
ArrayList<Cook> list = new ArrayList<>();  
list.add( new Cook1( "우주연" ));  
List.add( new Cook2( "백종원" ));  
List.add( new Cook3( "이연복" ));  
List.add( new KSJCook( "고든램지" ));
```

...

2. 전체요리사 목록을 3개의 그룹으로 나눈다

```
ArrayList<Cook> list = new ArrayList<>();  
ArrayList<한식가능한> 한식요리사= new ArrayList<>();  
ArrayList<중식가능한> 중식요리사 = new ArrayList<>();  
ArrayList<일식가능한> 일식요리사= new ArrayList<>();
```

```
for( Cook cook : list){  
    if( cook instanceof 한식가능한){  
        한식요리사.add ( cook);  
    }else if( cook instanceof 중식가능한){  
        중식요리사.add( cook);  
    }else if( cook instanceof 일식가능한){  
        일식요리사.add( cook);  
    }  
}
```

3. 컴퓨터가 랜덤수로 3개의 그룹 중 하나를 선택한다.

```
int random = (int) (Math.random() *3); //0 ,1, 2
```

```
String[] cookKind = { “한식가능한” , “중식가능한 ”, “일식가능한 ”};
```

```
String selectCook = cookKind[random];
```

```
System.out.println( selectCook + “선택되었어요 !! 기대Up ”);
```

4. 선택된 요리사 그룹에서 한 명의 요리사를 선택합니다

```
int size =0;  
switch(selectCook){  
    case “한식가능한” :  
        size = 한식요리사.size();  
        break;  
    case “중식가능한” :  
        size = 중식요리사.size();  
        break;  
    case “일식가능한” :  
        size = 일식요리사.size();  
}  
}
```

5. 그룹내의 랜덤으로 정한다

```
int randomOne =Math.floor( Math.random() * size );
String cookName="";
switch(selectCook ){
    case “한식가능한” :
        cookName= 한식요리사.get( randomOne);
        break;
    case “중식가능한” :
        cookName= 중식요리사.get( randomOne);
        break;
    case “일식가능한” :
        cookName= 일식요리사.get( randomOne);
}
}
```

6. 30초 후에 정해진 요리사클래스이름을 출력한다

```
try{  
    Thread.sleep( 30000 );  
    System.out.println( “결과 축하합니다” );  
    System.out.println( cookName );  
}  
catch( Exception e){  
    e.printStackTrace();  
}
```