

객체지향 프로그래밍 7장실습

2016112158김희수



[날짜]

[회사 이름]

[회사 주소]

이번 실습은 HashMap과 Vector를 이용해서 나라와 수도맞추기 게임을 구현해보는 것이다.

먼저 HashMap<String, String> nations = new HashMap<String,String>();으로 해시맵 객체를 생성한다. 그후 Scanner로 입력을 받아온다.

Vector의 경우 Vector<Nation> nations = new Vector<Nations>();로 Nation의 Vector를 생성해준다.

입력된 값이 1,2,3 중에 하나라면 case문을 이용해 각 경우에 따른 코드를 실행하고 아니라면 default로 지정해놓아 메뉴로 돌아간다.

입력된 값이 1인 경우, 나라와 수도를 입력한다. 현재 나라의 개수(current\_size)를 설정하고 나라와 수도를 입력받되 나라에 “그만”이라고 입력받으면 그만 입력하고 메뉴로 돌아간다. 단, 나라가 이미 해시맵에 존재한다면 입력한 나라는 이미 있다고 출력한다. 이 부분은 keySet()메소드를 이용해 구현하였다.

Vector로는 nations의 모든 원소에 대해 입력한 나라와 원소의 country가 같은지를 비교해야한다. 같다면 이미 있습니다라고 출력한다.

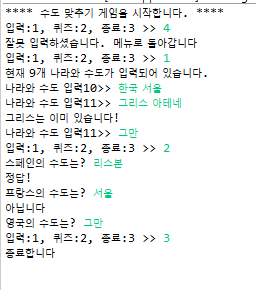
입력된 값이 2인 경우, 나라와 수도맞추기 게임을 시작한다. 먼저 해시맵의 경우를 보자. 나라들로 이루어진 리스트를 만들고 그 리스트에서 랜덤하게 나라를 뽑아오기 위해 섞는다. 이는 List와 Collections.shuffle()메소드로 구현하였다. 그 후 무작위적으로 뽑아온 나라의 수도를 입력받고 해당 나라의 수도가 맞으면 정답!, 아니면 아닙니다!를 출력한다. 이는 해시맵의 get()메소드로 구현하였다.

Vector도 비슷하다. 단, Vector의 경우 keySet()메소드나 get()메소드가 없으므로 조금 복잡해진다. 먼저 keys라는 List를 만들고. keys에 nations의 모든 원소의 country를 저장한다. 이제 무작위적으로 뽑기 위해 keys를 shuffle한다. 거기서 뽑은 key에 해당하는 nations의 capital를 출력하면 된다. 이를 위해, nations의 모든 원소에 대해 country를 temp\_country라고 임시로 저장하고 그 temp\_country와 key를 비교한다. 이 비교를 통해 해당 key값을 갖는 Nation객체의 인덱스를 찾을 수 있다. 이제 그 인덱스의 원소의 capital값과 answer를 비교하면 원하던 결과가 나온다.

입력된 값이 3인 경우, 종료한다. flag를 통해 while문을 빠져나간다.

만약에 해시맵 또는 Vector에 나라, 원소가 한 개도 없을 때 메뉴에서 2를 입력한다면 IndexOutOfBoundsException이 발생하므로 try-catch문을 이용해 예외를 처리해준다.

<미리 해시맵(또는 Vector)에 원소를 미리 삽입해놓은 경우>



<해시맵(또는 Vector)에 원소가 하나도 없을 경우>

