

다음과 같은 프로그램을 DOM API를 이용하여 작성하시오. (DOMLab1 예제 활용)

1. mondial.xml에서 각 국가(country)의 이름, 면적, 인구, 수도(capital)의 이름을 찾아 console에 출력한다.

예: Albania

면적: 28750

인구: 3249136

수도의 이름: Tirane

...

is-country-cap = "yes"
capital

2. 각 국가에 대해 이름, 면적, 인구, 수도의 정보를 선택하여 다음과 같은 형식의 XML 문서로 저장한다. (기존 DOM 트리를 수정한 후 새로운 문서로 저장함. result-sample.xml 참조)

```
<mondial>
  <country>                                <!-- <country>의 모든 속성들을 삭제함 -->
    <name>Albania</name>
    <code>AL</code>                        <!-- 국가 코드를 나타내는 <code>를 <name> 다음에 추가함 -->
    <population>3249136</population>
    <city id="cty-cid-cia-Albania-Tirane">    <!-- 수도(capital)에 해당하는 엘리먼트 -->
      <name>Tirane</name>                    <!-- 수도의 id 속성과 자손들을 모두 포함 -->
      <longitude>19.8</longitude>
      <latitude>41.3</latitude>
      <population year="87">192000</population>
    </city>
    <!-- <country>에 관한 다른 정보(child element)들은 모두 삭제함 -->
  </country>
  ...                                       <!-- <country>들 외의 다른 정보(sibling element)들은 모두 삭제함 -->
</mondial>
```

3. 각 국가의 인구 정보를 이용하여 대륙별 인구를 계산 및 출력한다.

- 각 국가가 속한 대륙은 <encompassed> 엘리먼트의 continent 속성을 참조한다. 여러 대륙에 걸쳐 있는 국가(예: 터키, 러시아 등)의 경우, 가장 많이 속한 대륙에 그 나라의 전체 인구가 속한다고 가정한다 (예: 터키는 아시아에 95% 속해 있으므로 터키의 인구는 모두 아시아에 속하는 것으로 계산함). 국가의 대륙별 소속 비율을 <encompassed>의 percentage 속성을 통해 알 수 있다.
- 인구의 합은 매우 큰 숫자가 될 수 있으므로 long 타입 변수를 이용하여 저장한다. 문자열을 long 값으로 바꾸려면 Long.parseLong(String s) 메소드를 이용한다.
- long 타입 값을 출력할 때 세 자리 마다 comma를 찍으려면 다음과 같이 java.text.NumberFormat 클래스의 format() 메소드를 이용하여 문자열로 변환한다.

```
NumberFormat formatter = NumberFormat.getInstance();
formatter.format(longNumber);
```

- 실행 예:

Populations of the Continents

Europe의 인구: 580,580,439명

Asia의 인구: 3,648,685,183명

America의 인구: 784,188,664명

Africa의 인구: 731,161,250명

Australia의 인구: 29,833,722명

합계: 5,774,449,258명

4. (선택 문제) 각 국가의 인구와 종교 정보를 이용하여 전 세계의 종교별 신자 수를 계산 및 출력한다.

- 각 국가의 종교는 <religions> 엘리먼트의 값을 참조하고, percentage 속성을 통해 종교의 인구 비율을 알 수 있다. 이 비율을 고려하여 각 국가의 종교별 신자 수를 계산하고, 각 종교별로 모든 국가에 속한 신자 수를 합산한다.
- 종교들의 이름이나 개수는 미리 가정하지 말고 mondial 문서에서 알아내야 한다. 이것을 사전에 처리하거나, 또는 각 국가의 정보를 처리하면서 새로운 종교가 발견되는 대로 처리한다.
- 종교별 신자 수를 저장하기 위해 배열이나 List 또는 Map을 이용한다. List 보다 Map을 이용하면 (종교 이름, 신자 수) 쌍을 효율적으로 저장 및 활용할 수 있다.
- 결과 출력 시 신자 수가 많은 종교부터 내림차순으로 정렬하여 출력할 수 있도록 구현해 본다.
 - Hint: 종교 이름과 신자 수를 함께 저장하는 별도의 클래스와 그 객체들을 신자 수를 기준으로 정렬하기 위한 Comparator 클래스를 정의하고, Arrays.sort(...) 또는 Collections.sort(...) 메소드를 활용하면 효율적으로 구현 가능

주의사항

- 구현 과정에서는 mondial-sample.xml 파일을 이용해서 테스트하고, 프로그램이 완성된 후에는 mondial.xml을 이용하여 최종 결과를 생성함
- 소스코드 내에 주석을 달아 간략히 설명할 것

제출방법: 아래의 내용을 과제 게시글 안에 직접 copy & paste하여 제출함

- main()을 제외한 나머지 메소드들의 소스 코드
- 프로그램 실행 결과 (내용이 많을 경우 중간 부분 생략)