

Langages pour le Web

☐0 ☐0
☐1 ☐1
☐2 ☐2
☐3 ☐3
☐4 ☐4
☐5 ☐5
☐6 ☐6
☐7 ☐7
☐8 ☐8
☐9 ☐9

L3 MIAGE
17 Décembre 2020

Durée: 2 heures
Barème sur 22 points

Une feuille de notes est autorisée
Certaines réponses doivent être données sur le sujet.
Vous êtes autorisés à répondre sur feuille séparée, mais
le sujet, même non rempli, doit être rendu
intégralement !

Ne pas mettre son nom sur le sujet, ni son numéro
d'étudiant à gauche (place réservée au numéro de copie).

La mairie de Villard de Lans, petite ville du massif du Vercors et station de sports toutes saisons, souhaite mettre en place un flux d'informations RSS informant sur la météorologie, à raison de deux relevés par jour, un à 6h00 du matin, l'autre à midi. Ces flux doivent renseigner sur 3 points : (1) *la température* incluant mesure de température (une température est une valeur supérieure à -273°C) et éventuellement la valeur de l'isotherme zéro (altitude en m à laquelle il fait 0°C , cette mesure n'étant pas nécessairement fournie), (2) *les précipitations* incluant le niveau de précipitations en mm d'eau et le type de précipitations (*aucune, pluie, grésil, grêle, neige*), et (3) *l'anémométrie*, c'est à dire la vitesse du vent en km/h , avec éventuellement la vitesse des rafales de vent (cette mesure n'étant pas nécessairement fournie), et la direction du vent avec 8 directions possibles (*N, NO, O, SO, S, SE, E, NE*), cette information n'étant pas nécessairement fournie (par exemple lorsqu'il n'y a pas de vent). Les données sont stockées au format XML, selon un schéma suffisamment générique pour être par la suite étendu à la communauté de communes de ce massif montagneux. Un fichier par mois est produit et est nommé *meteo-ville-mmyyyy.xml* (par exemple *meteo-VillarddeLans-122017.xml*). En plus des bulletins météo, chaque fichier contient le renseignement sur la période de l'année (mois et année), ainsi que les informations sur le site sur lequel sont effectuées les mesures. Les informations géographiques sont au format KML. *NB*: Tous les renseignements nécessaires à la modélisation se trouvent dans ce texte !

1 Modélisation des données météo en XML et XMLSchema (7 points)

Dans cette partie, nous allons modéliser les données météorologiques convenablement. On fournit ci-dessous une instance de fichier XML tel que souhaité pour le mois de décembre ; on ne montre ici que les bulletins du 18 décembre 2017.

→ Fichier *meteo-VillarddeLans-122017.xml* :

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <met:météo
3   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
4   xmlns:kml='http://www.opengis.net/kml/2.2'
5   xmlns:gx="http://www.google.com/kml/ext/2.2"
6   xmlns:met='http://www.meteo-france.fr/bulletin'
7   xsi:schemaLocation='http://www.meteo-france.fr/bulletin ./meteo.xsd'
8   url="http://www.espace-villard-correncon.fr/meteo-villard-de-lans.htm">
9   <met:période année="2017" mois="12"/>
10  <met:lieu>
11    <kml:name>Villard de Lans</kml:name>
12    <kml:description>Alpes Vercors Nord</kml:description>
13    <kml:Point>
14      <gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode>
15      <kml:coordinates>5.550886412795313,45.06638495609347,1008</kml:coordinates>
16    </kml:Point>
17  </met:lieu>
18  <!-- Bulletins d'exemple -->
19  <met:bulletin date="2017-12-16T06:00:00+01:00">
20    <met:température>-6.5</met:température>
21    <met:anémométrie>

```

CORRECTION

```

22         <met:vitesse>10</met:vitesse>
23     </met:anémométrie>
24     <met:précipitations type="neige">
25         <met:niveau>15.0</met:niveau>
26     </met:précipitations>
27 </met:bulletin>
28 <met:bulletin date="2017-12-16T12:00:00+01:00">
29     <met:température isothermeZero="600">-2.0</met:température>
30     <met:anémométrie direction="NE">
31         <met:vitesse>30</met:vitesse>
32         <met:rafales>55</met:rafales>
33     </met:anémométrie>
34     <met:précipitations type="aucune"/>
35 </met:bulletin>
36 <!--
37     A COMPLETER : Bulletins du 21/12/17
38 -->
39 </met:météo>

```

On donne également le squelette du schéma XML correspondant:

→ Fichier *meteo.xsd* :

```

1  <?xml version="1.0"?>
2  <xs:schema version="1.0"
3      xmlns:xs="http://www.w3.org/2001/XMLSchema"
4      xmlns:met="http://www.meteo-france.fr/bulletin"
5      xmlns:kml="http://www.opengis.net/kml/2.2"
6      targetNamespace="http://www.meteo-france.fr/bulletin"
7      elementFormDefault="qualified">
8
9      <xs:import namespace="http://www.opengis.net/kml/2.2"
10         schemaLocation="http://schemas.opengis.net/kml/2.2.0/ogckml22.xsd"/>
11      <xs:import namespace="http://www.google.com/kml/ext/2.2"
12         schemaLocation="http://code.google.com/apis/kml/schema/kml22gx.xsd"/>
13
14      <!-- élément racine -->
15      <!-- Type Météo -->
16      <!-- Type Période -->
17      <!-- Type Bulletin -->
18      <!-- Type ValeurTempérature -->
19      <!-- Type Température -->
20      <!-- Type Anémométrie -->
21      <!-- Type Direction -->
22      <!-- Type Précipitations -->
23      <!-- Type TypePrécipitation -->
24      <!-- Type DoublePositif -->
25
26  </xs:schema>

```

Question 1 Dans l'espace réservé ci-dessous, modélisez les types (i) *TypePrécipitation*, (ii) *ValeurTempérature* et (iii) *Direction* qui représentent respectivement (i) les différents types de précipitations, (ii) les valeurs que peuvent prendre la température et (iii) les différentes directions du vent. Ecrivez également un type *DoublePositif* qui sera utilisé pour les valeurs de vitesse du vent et les niveaux de précipitations (positives). Ecrivez enfin un type *Période* dont les attributs *année* et *mois* correspondent aux types XML Schéma standard *xs:gYear* et *xs:gMonth*.

..... ☒ A ☐ B ☐ C ☐ D ☐ E *Réservé au correcteur*

Question 2 Sur feuille séparée, modélisez les types *Précipitations*, *Température* et *Anémométrie* (vitesse du vent) de telle manière que cela corresponde à l'instance de document et aux renseignements donnés dans le texte. Ecrivez également le type *Bulletin* et le type *Météo*. Ecrivez enfin l'élément racine.

NB: Le type XML Schéma standard de date employé, *xs:dateTime*, encode la date, l'heure et le fuseau horaire ; la date et l'heure sont séparés par la lettre 'T'. Le lieu est de type *kml:PlacemarkType* du langage KML.

..... ☒ A ☐ B ☐ C ☐ D ☐ E *Réservé au correcteur*

2 Transformation XSLT vers des documents RSS(7 points)

On souhaite transformer chaque jour les bulletins du jour au format RSS comme dans le document ci-dessous.

→ Fichier *meteo.rss* :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rss version="2.0"
3   xmlns:kml="http://www.opengis.net/kml/2.2"
4   xmlns:met="http://www.meteo-france.fr/bulletin">
5   <channel>
6     <title>Météo</title>
7     <date>2017-12-14</date>
8     <link>Météo des neiges [Villard de Lans] : http://www.espace-villard-correncon.fr/meteo
9       -villard-de-lans.htm</link>
10    <description>Bulletins météo du 2017-12-14 pour Villard de Lans, secteur Alpes Vercors
11      Nord</description>
12    <item>
13      <title>Bulletin météo du matin</title>
14      <pubDate>2017-12-14 - 06:00:00 - GMT+01:00</pubDate>
15      <description>
16        Température : -6.5 degrés
17        Précipitations [neige]: 15.0 mm
18        Vent : 05 km/h
19      </description>
20    </item>
21    <item>
22      <title>Bulletin météo de l'après-midi</title>
23      <pubDate>2017-12-14 - 12:00:00 - GMT+01:00</pubDate>
24      <description>
25        Température : 2.0 degrés
26        Précipitations [aucune]
27        Vent : 30 km/h avec rafales à 55 km/h
28      </description>
29    </item>
30  </channel>
31 </rss>
```

Question 3 Complétez la feuille de transformation XSLT de façon à obtenir un fichier RSS semblable au document d'exemple donné ci-dessus. La date sélectionnée est transmise par paramètre via l'application. La feuille de transformation doit permettre notamment l'affichage de 1 à 2 bulletins par jour, selon disponibilité. Elle ne doit évidemment comporter qu'un seul template pour les deux bulletins. *NB*: vous pourriez avoir besoin de la fonction XPath `substring(s:string,start:int,offset:int)`. Rappel: la structure de branchement if/else en XSLT s'obtient avec `<xsl:choose><xsl:when test="....">....</xsl:when><xsl:otherwise>....</xsl:otherwise></xsl:choose>`.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3     .....
4     .....
5     .....
6     >
7     <!-- Sortie de la transformation -->
8     <xsl:output method="....." version="1.0" encoding="iso-8859-15"
9         omit-xml-declaration="no"/>
10    <!-- Paramètre date transmis par l'application -->
11    <xsl:param name="date"/>
12    <!-- Template principal -->
13    <xsl:template match="....">
14
15
16
17
18
19
20
21
22
23
24    </xsl:template>
25    <!-- Pour afficher chaque Bulletin -->
26    <xsl:template match=".....">
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47    </xsl:template>
48 </xsl:stylesheet>

```

3 Lecture SAX et affichage Java du document RSS(8 points)

Une application Java est proposée pour lire les flux RSS. On s'intéresse ici à la lecture du document par un parseur SAX et à son affichage. On souhaite que le fichier RSS, *meteo.rss*, soit lu et mis en forme par l'application Java comme suit : → Fichier *exempleMiseEnFormeRSS.txt* :

```
----- Bulletin Météo -----
Villard de Lans, le 2017-12-14
Matin : Température : -0.5°C, Précipitations [aucune], Vent : 0 km/h
Après-midi : Température : 3.0°C, Précipitations [aucune], Vent : 15 km/h
-----
```

→ Fichier *MeteoSAX.java* à compléter :

```
1 import org.xml.sax.Attributes;
2 import org.xml.sax.helpers.DefaultHandler;
3
4 public class MeteoSAX extends DefaultHandler {
5
6     private String meteoText;
7     private enum ETAT {RSS, DATE, ITEMS, DESCRIPTION}
8     private enum MOMENT {MATIN, APRESMIDI}
9     private ETAT etat;
10    private MOMENT moment;
11
12    @Override
13    public void startDocument() {
14        etat = .....(1);
15        moment = .....(2);
16        meteoText = "----- Bulletin Météo
17                    -----\n";
18        meteoText += .....(3);
19    }
20
21    @Override
22    public void startElement(String namespaceURI, String localName, String
23        qName, Attributes atts) {
24        switch (etat) {
25            case RSS:
26                if (localName.equals( .....(4) )) { etat = ETAT.DATE;
27                }
28                if (localName.equals("item")) { etat = .....(5) ; }
29                break;
30            case ITEMS:
31                if (localName.equals( .....(6) )) {
32                    etat = .....(7);
33                    switch (moment) {
34                        case .....(8) :
35                            .....(9)
36                            break;
37                        case .....(8) :
38                            .....(11)
39                            break;
40                    }
41                }
42            }
43        }
44
45    @Override
46    public void characters(char[] ch, int start, int length) {
47        if ( .....(12) || .....(13) ) {
48            for (int i = start; i < start + length; i++) {
49                meteoText += ch[i];
50            }
51            meteoText += .....(14) ;
52        }
53    }
```

CORRECTION

```

51
52  @Override
53  public void endElement(String namespaceURI, String localName, String
54      qName) {
55      switch (etat) {
56          case DATE:
57              etat = .....(15) ;
58              break;
59          case DESCRIPTION:
60              if (localName.equals("description")) {
61                  if (moment == .....(16) ) {
62                      etat = .....(17) ;
63                      moment = .....(18) ;
64                  } else {
65                      meteoText += "
66                      -----\n";
67                      etat = .....(19) ;
68                  }
69              }
70              break;
71      }
72  }
73  public String getText() {
74      return meteoText;
75  }
76  }

```

Question 4 Complétez le document ci-dessus (points 1 à 19) de façon à ce que cela génère le texte montré en exemple dans le fichier exempleMiseEnFormeRSS.txt ci-dessus.

..... ☒ A ☐ B ☐ C ☐ D ☐ E *Réservé au correcteur*