Exercise 5: Courses and Participants

Given a text file that describes courses and their participants. Each course has a name and a list of participants according to the following grammar:

```
Courses       = {Course}.
Course        = "course" CourseName ":" Participants.
Participants  = {Participant}.
CourseName = string.
Participant   = string.
```

Turn it into an attributed grammar, which builds a global hash table that contains for each participant the list of attended courses. Use the following global data structure:

```
HashMap<String, ArrayList<String>> participants;
```

At the end, print this list. Take the value of the terminal symbol `string` from `t.val`.

Solution

The first production processes the global hash map `participants` and prints for each participant the list of attended courses:

```
Courses
= { Course }              (. for (String name: participants.keySet()) {
                               System.out.print(name + ": ");
                               boolean first = true;
                               for (String course: participants.get(name)) {
                                   if (first) first = false; else System.out.print(", ");
                                   System.out.print(course);
                               }
                               System.out.println();
                            } .).
```

The second production obtains a course name from `CourseName` ans simply passes it on to `Participants`.

```
Course                    (. String course; .)
=  "course" CourseName <↑course> ":" Participants <↓course>.
```

The production of `Participants` enters the course name into the course list of each participant.

```
Participants <↓course>      (.  String course, name; .)
=  {   Participant <↑name>    (.  ArrayList<String> courses = participants.get(name);
                                 if (courses == null) {
                                     courses = new ArrayList<String>();
                                     participants.put(name, courses);
                                 }
                                 courses.add(course); .)
   }.
```

The productions of `CourseName` and `Participant` finally provide the values of strings without quotes.

```
CourseName <↑course>    (. String course; .)
= string                (. course = t.val.substring(1, t.val.length()-1); .).

Participant <↑name>     (. String name; .)
= string                (. name = t.val.substring(1, t.val.length()-1); .).
```