******Conference******

```
/* Classe Conference */
/* Doit inclure les attributs d'une conference : son nom, ses sessions et ses
    participants */
/* Doit inclure les methodes d'acces pour ses attributs et son contructeur */
/* Surcharge de la methode toString() pour affichage */

package conf;

import java.lang.String;
import java.util.*;

public class Conference {
        public String nom;
        public Collection<Session> sessions = new HashSet<Session>(); // of
            type Session
        public Collection<Participant> participants = new HashSet<Participant
            >(); // of type Participant


        public String getNom() {
                return nom;
        }

        public void setNom(String nom) {
                this.nom = nom;
        }

        public java.util.Collection getParticipant() {
                return participants;
        }

        public void setParticipant(Collection<Participant> participants) {
                this.participants = participants;
        }

        public java.util.Collection getSession() {
                return sessions;
        }

        public void setSession(Collection<Session> sessions) {
                this.sessions = sessions;
        }

        public void addParticipant(Participant p){
                if (! this.participants.contains(p)) this.participants.add(p)
                    ;
        }

        public void addSession(Session s){
                if (! this.sessions.contains(s)) this.sessions.add(s);
        }

        public Conference(String nom) {
                this.nom = nom;
        }
```

```
        public String toString(){
                String s = new String();
                s = nom +"\nParticipants : \n";
                Iterator i1 = participants.iterator();
                while (i1.hasNext())
                        s=s+"\t"+((Participant) i1.next()).toString()+"\n";
                s=s+"\nSessions :\n";
                Iterator i2 = sessions.iterator();
                while (i2.hasNext())
                        s=s+"\t\t"+((Session) i2.next()).toString()+"\n";
                return s;
        }

}
```

******Participant******

```
/* Classe Participant */
/* Doit inclure les attributs d'un participant (simplifies dans cette version
   ) nom, prenom, adresse et organisation */
/* Doit inclure les methodes d'acces pour un participant, son constructeur */
/* Surcharge de la methode toString() pour affichage */

package conf;

public class Participant {
        private String nom;
        private String prenom;
        private String adresse;
        private String organisation;

        public String getAdresse() {
                return adresse;
        }
        public void setAdresse(String adresse) {
                this.adresse = adresse;
        }
        public String getNom() {
                return nom;
        }
        public void setNom(String nom) {
                this.nom = nom;
        }
        public String getOrganisation() {
                return organisation;
        }
        public void setOrganisation(String organisation) {
                this.organisation = organisation;
        }
        public String getPrenom() {
                return prenom;
        }
        public void setPrenom(String prenom) {
                this.prenom = prenom;
        }

        public Participant(String nom, String prenom, String adresse, String
           organisation) {
                super();
                this.nom = nom;
                this.prenom = prenom;
                this.adresse = adresse;
                this.organisation = organisation;
        }

        public String toString() {
                return getNom()+"\t"+getPrenom()+"\t"+getAdresse()+"\t"+
                   getOrganisation();
        }

}
```

******Conferencier******

```
/* Classe Conferencier */
/* Herite de la classe Participant */
/* Doit inclure les attributs d'un conferencier : nom de la session pour
   laquelle il est orateur, et son numero de passage */
/* Doit inclure les methodes d'acces pour un conferencier, ses constructeurs
   */
/* Surcharge de la methode toString() pour affichage */
/* Surcharge des methodes de comparaison compareTo : OBLIGATOIRE pour
   implementer Comparable, utilis  pour le TreeSet dans Session*/


package conf;


public class Conferencier extends Participant implements Comparable{

        private String nomSession;
        private int numero;


        public Conferencier(String nom, String prenom, String
            formulePolitesse,
        /* Constructeur avec simplement les champs de la superclasse
            Participant */
                        String organisation) {
                super(nom, prenom, formulePolitesse, organisation);
        }

        public Conferencier(String nom, String prenom, String
            formulePolitesse, String organisation, String nomSession) {
        /* Constructeur avec les champs de la superclasse Participant ET le
            champ du nom de la session */
                super(nom, prenom, formulePolitesse, organisation);
                this.nomSession = nomSession;
        }

        public Conferencier(Participant p,String nomSession){
                /* Constructeur avec un participant de la superclasse
                    Participant ET le champ du nom de la session */
                super(p.getNom(),p.getPrenom(),p.getAdresse(),p.
                    getOrganisation());
                this.nomSession = nomSession;
        }

        public String getNomSession() {
                return nomSession;
        }

        public void setNomSession(String nomSession) {
                this.nomSession = nomSession;
        }

        public int getNumero() {
                return numero;
        }
```

```
        public void setNumero(int numero) {
                this.numero = numero;
        }


        public String toString(){
                return super.toString()+"\t"+nomSession+"\tordre = "+numero;
                //return "conferencier";
        }

        /* Cette m thode est OBLIGATOIRE pour faire un TreeSet de
           Conferencier*/
        /* Elle implemente la classe Comparable */
        public int compareTo(Object o){
                 int n2 = ((Conferencier) o).getNumero();
                 int n1 = this.getNumero();
                 return n1==n2 ? 0 : (n1<n2 ? -1 : 1);

        }
/*      Peut etre utilisee pour tester l'egalite entre instances de
    Conferencier*/
        public boolean equals(Object o){
                  if (!(o instanceof Conferencier)) {
                      return false;
                    }
                  int n2 = ((Conferencier) o).getNumero();
                  int n1 = this.getNumero();
                 return (n1==n2);
        }

}
```

******Session******

```
/* Classe Session */
/* Doit inclure les attributs d'une session : son nom et ses conferenciers */
/* Doit inclure les methodes d'acces pour une session et son constructeur */
/* Definition de la methode d'ajout d'un conferencier a une session */
/* Surcharge de la methode toString() pour affichage */

package conf;

import java.util.*;

public class Session {
    private String nom;
        /* Grace a la definition de compareTo dans la classe Conferencier, le
            tri se fait automatiquement dans TreeSet*/
    public Collection<Conferencier> conferenciers = new TreeSet<Conferencier
        >();

    public Session(String nom) {
            this.nom = nom;
        }

        public Collection getconferenciers() {
            return conferenciers;
        }

        public void setconferenciers(Collection<Conferencier> conferenciers)
            {
            this.conferenciers = conferenciers;
        }


        public String getNom() {
            return nom;
        }
        public void setNom(String nom) {
            this.nom = nom;
        }
        public Session(String nom, Collection<Conferencier> conferenciers) {
            this.nom = nom;
            this.conferenciers = conferenciers;
        }

        /*Cette methode est utilisee pour savoir quel est le max de l'ordre
            de passage */
        public int numeroCourant(){
            Iterator i = this.conferenciers.iterator();
            int max = 0,courant;
            while (i.hasNext()){
                    courant = ((Conferencier) i.next()).getNumero();
                    if (courant>max) max = courant;
            }
            return max;
        }

        /* Cette methode est utilisee pour affecter automatiquement un numero
            a un conferencier : */
```

```
        /* on attribut au dernier conferencier ajoute, le max de l'ordre de
           passage + 1 */
        public void addConferencierNum(Conferencier c){
                if (! this.conferenciers.contains(c)) {
                        c.setNumero(numeroCourant()+1);
                        this.conferenciers.add(c);
                        }
        }

        public void addConferencier(Conferencier c){
                        this.conferenciers.add(c);
                        }

    public String toString(){
        String s = new String();
        s=nom+"\n";
        Iterator i = conferenciers.iterator();
        while (i.hasNext()) {
                s=s+"\t"+((Conferencier) i.next()).toString()+"\n";
        }
        return s;
    }
}
```

******Test******

```
/* Classe Test */
/* Ni attributs ni methodes, simplement le main*/
/* Utilise les classes Conference, Paticipant, Conferencier, Session*/

package conf;

public class Test {

	public static void main(String[] args) {

	/* Creation des participants ou conferenciers */
	Conferencier c1 = new Conferencier("toto","titi","Toulouse","irit","
	    session1");
	Conferencier c2 = new Conferencier("tata","tutu","Toulouse","irit","
	    session1");
	Participant p3 = new Participant("Anne","B","Bordeaux","labri");
	Conferencier c4 = new Conferencier("Benoit","C","Bordeaux","labri","
	    session2");
	Conferencier c5 = new Conferencier("p5","n5","a5\t","o5","session1");
	Conferencier c6 = new Conferencier("p6","n6","a6\t","o6","session2");
	Conferencier c7 = new Conferencier("p7","n7","a7\t","o7","session3");
	Conferencier c8 = new Conferencier("p8","n8","a8\t","o8","session3");


	/* Creation de la conference */
	Conference conf = new Conference("visio");

	/* Ajout des participants/conferenciers a la conference */
	conf.addParticipant(c1);
	conf.addParticipant(c2);
	conf.addParticipant(p3);
	conf.addParticipant(c4);
	conf.addParticipant(c5);
	conf.addParticipant(c6);
	conf.addParticipant(c7);
	conf.addParticipant(c8);

	/* Creation de la session session1, mise a jour de l'ordre de passage
	    , et ajout des conferencier dans la session*/
	Session s1 = new Session("session1");
	c1.setNumero(2);
	s1.addConferencier(c1);
	c2.setNumero(1);
	s1.addConferencier(c2);
	c5.setNumero(3);
	s1.addConferencier(c5);

	/* Ajout de la session dans la conference */
	conf.addSession(s1);

	/* Creation de la session session2, mise a jour de l'ordre de passage
	    , et ajout des conferencier dans la session*/
	Session s2 = new Session("session2");
	c4.setNumero(2);
	s2.addConferencier(c4);
	c6.setNumero(3);
```

```
        s2.addConferencier(c6);
        conf.addSession(s2);

        /* Creation de la session session2, ajout des conferencier dans la
           session, ordre de passage donn  automatiquement*/
        Session s3 = new Session("session3");
        s3.addConferencierNum(c7);
        s3.addConferencierNum(c8);
        conf.addSession(s3);


        /* Affichage de la conference */
        System.out.println(conf);
        }

}
```

******OUTPUT******

```
visio
Participants :
        tata      tutu      Toulouse        irit      session1        ordre = 1
        toto      titi      Toulouse        irit      session1        ordre = 2
        Benoit    C         Bordeaux        labri     session2        ordre = 2
        p8        n8        a8              o8        session3        ordre = 2
        p7        n7        a7              o7        session3        ordre = 3
        p6        n6        a6              o6        session2        ordre = 3
        p5        n5        a5              o5        session1        ordre = 3
        Anne      B         Bordeaux        labri

Sessions :
                session1
        tata      tutu      Toulouse        irit      session1        ordre = 1
        toto      titi      Toulouse        irit      session1        ordre = 2
        p5        n5        a5              o5        session1        ordre = 3

                session2
        Benoit    C         Bordeaux        labri     session2        ordre = 2
        p6        n6        a6              o6        session2        ordre = 3

                session3
        p7        n7        a7              o7        session3        ordre = 3
        p8        n8        a8              o8        session3        ordre = 2
```