

Class Utilisateur

```
1     abstract class Utilisateur
2     {
3         private string login;
4         private string mdp;
5         private string role;
6
7         public Utilisateur(string L, string MdP){
8             this.login=L;
9             this.mdp=MdP;
10            role=Identifie();
11            if (role==null){
12                throw new Exception(String.Format("{0},{1}",login,mdp));
13            }
14        }
15        public override string ToString()
16        {
17            return String.Format("Utilisateur [{0},{1},{2}]", login, mdp, role);
18        }
19        abstract public string Identifie();
20    }
```

21 Class Admin

```
1     class Admin:Utilisateur
2     {
3         public Admin(string L, string MdP) : base(L, MdP) { }
4
5         public override string Identifie()
6         { //authentification LDAP
7             return "admin";
8         }
9     }
```

10 Class Observateur

```
1     class Observateur: Utilisateur
2     {
3         public Observateur(string L, string MdP) : base(L, MdP) { }
4         public override string Identifie()
5         { //authentification SGBD
6             return "Observateur";
7         }
8     }
```

9 Class Inconnu

```
1     class Inconnu : Utilisateur
2     {
3         public Inconnu(string L, string MdP) : base(L, MdP) { }
4         public override string Identifie()
5         {
6             return null;
7         }
8     }
9
10
```