



**OFPPT**

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle  
et de la Promotion du Travail

Direction Recherche et Ingénierie de la Formation

**Examen de fin de formation, session Juin 2011- CORRIGE**

Filière : **Techniques de Développement Informatique**  
Niveau : **TS**  
Durée : **4 heures**

Epreuve : **Théorique**

Barème : / **40**

### **Dossier 1 : Programmation (10 pts)**

```
Class DeveloppeurExterne
{
    Private Integer Matricule ;
    Private String Nom,Email ;
    Private float Taux ;
    public DeveloppeurExterne(Integer _Matricule,String _Nom, String _Email, float _Taux)
    {
        Matriule=_Matricule ;
        Nom=_Nom ;
        Email=_Email ;
        Taux=_Taux ;
    }
    Public Integer Matricule()
    { return Matricule ; }
    Public void Matricule(Integer _Matricule)
    { Matricule=_Matricule ; }
    Public String Nom()
    { return Nom ; }
    Public void Nom(String _Nom)
    { Nom=_Nom ; }
    public String Email()
    { return Email ; }
    public void Nom(String _Email)
    { Email =_Email ; }
    public float Taux()
    { return Taux ;}
    public void Nom(String _Taux)
    { Taux =_Taux ; }
    float Calcul_Salaire(int Hs)
    { return Hs*Taux;}
}
```

public DeveloppeurInterne extends DeveloppeurExterne

```

{ String Catégorie ;
public DeveloppeurInterne(Integer _Matricule,String _Nom, String _Email,String _Catégorie)
{
    Super(_Matricule, _Nom, _Email, _Taux ) ;
    Catégorie=_Catégorie ;
    if (Catégorie=="S") Taux=200 ;
    else Taux=150;
}
public float Catégorie()
{ return Catégorie ;}
public void Catégorie(String _Catégorie)
{ Catégorie =_Catégorie ; }

```

// Redéfinition d'une méthode héritée

```

float Calcul_Salaire(int Hs)
{ float Salaire;
  if (Catégorie=="S") Salaire=8000 ;
  else Salaire=5000;
  Salaire+=Salaire+super.Calcul_Salaire(Hs) ;
  return salaire ;
}

```

Class projet

```

{ Private Int Id ;
  Private String Intitule ;
  Private Hashtable liste ;
  projet(int _Id ,String _Intitule)
  { Id=_Id ;
    Intitule=_Intitule ;
    Liste=new Hashtable() ;
  }
  void AjoutParticipant(DeveloppeurExterne D)
  { liste.put(D.Matricule() ,D) ;
  }
  float CalSalaireTousParticipant()
  { float salaire ;
    Enumeration e = liste.elements();
    while ( e.hasMoreElements())
    { salaire += salaire + ((DeveloppeurExterne)e.nextElement()).Calcul_Salaire();
    }
    return salaire ;
  }
}

```

Program principal :

```

DeveloppeurExterne D1=new DeveloppeurExterne(1000 ""ALAMI" , "Alami@yahoo.fr " , 250 ) ;
DeveloppeurExterne D2=new DeveloppeurExterne(1000 ""SEFRIOUT" , "Sef@caramail.fr " , 320 ) ;
DeveloppeurExterne D3=new DeveloppeurInterne(1000 "BELLAM" , "bel@hotmail.fr " , "J") ;

```

```

Projet P1=new Projet(123, "Projet123 ") ;

```

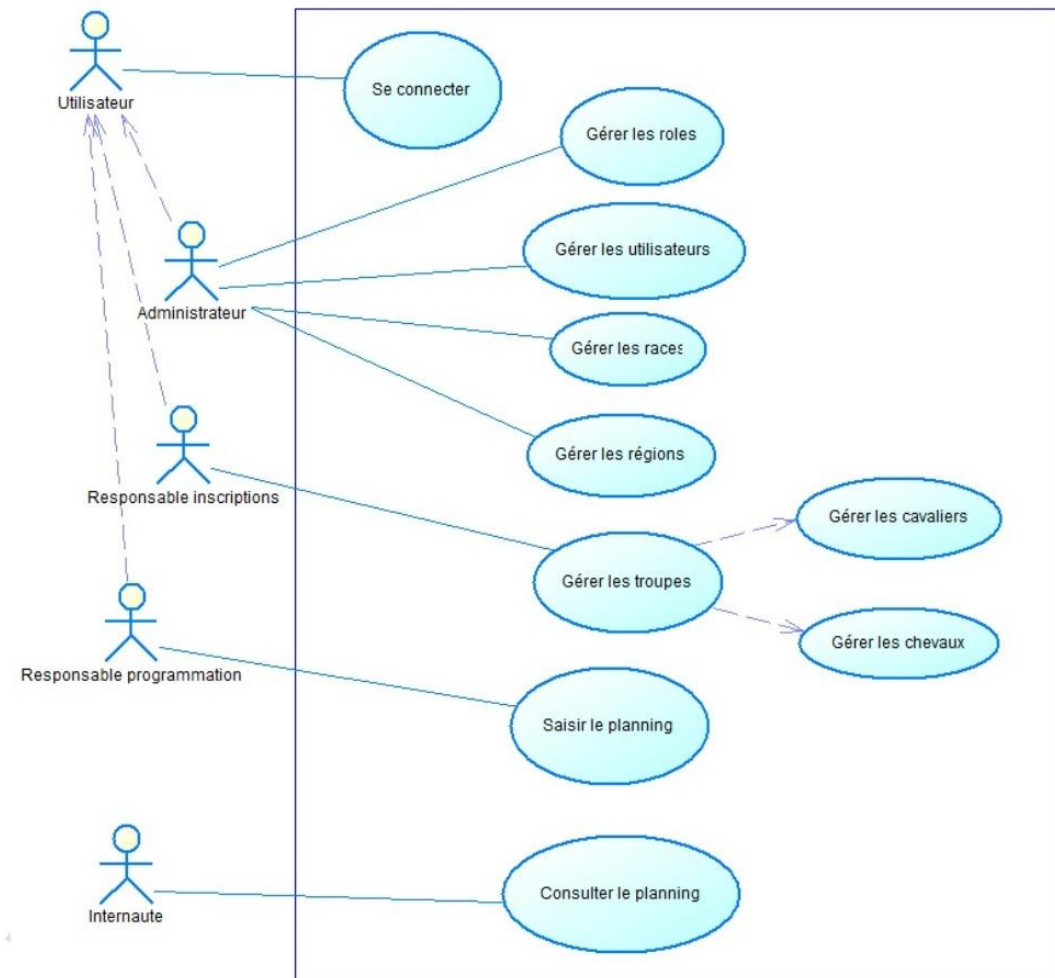
```

P1.AjoutParticipant(D1) ;
P1.AjoutParticipant(D2) ;
P1.AjoutParticipant(D3) ;
System.out.println("Le salaire global de tous les participant au projet est "+P.
CalSalaireTousParticipant());

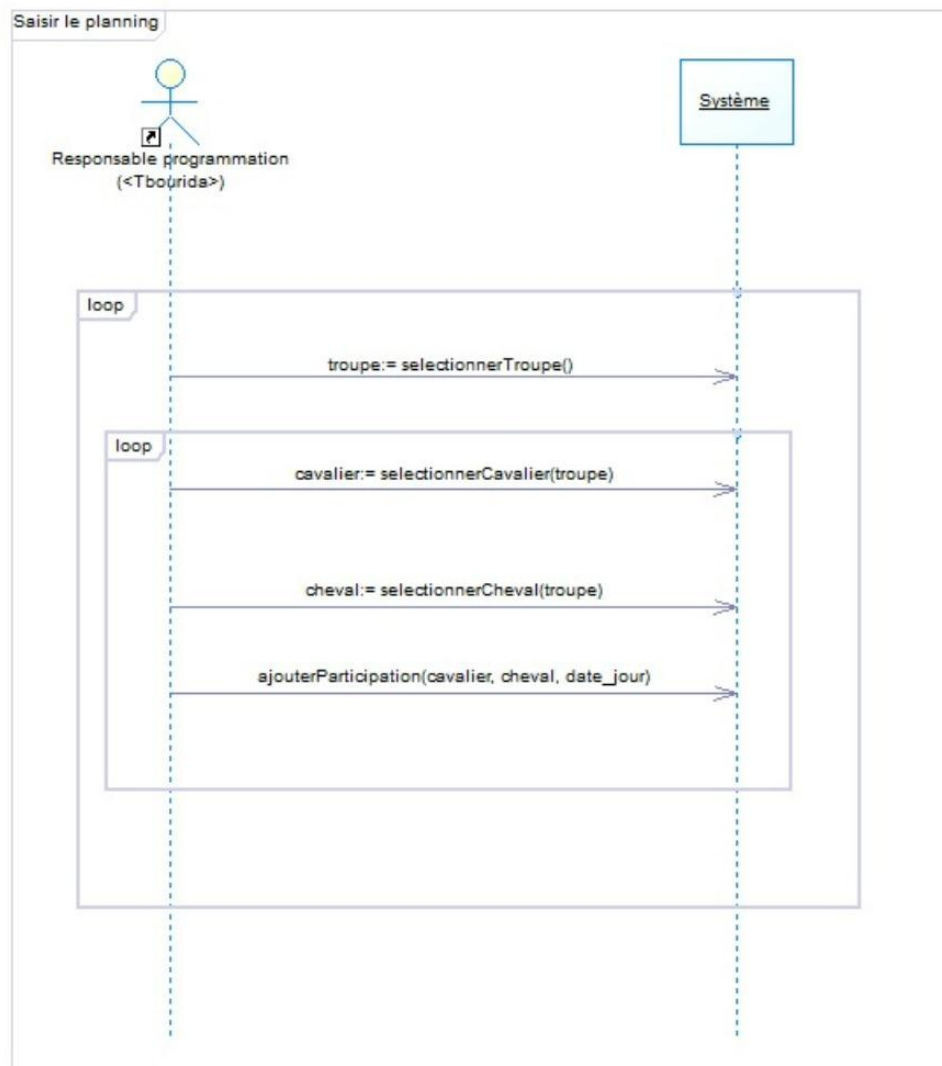
```

## **Dossier 2 : Modélisation orientée objet (14 pts)**

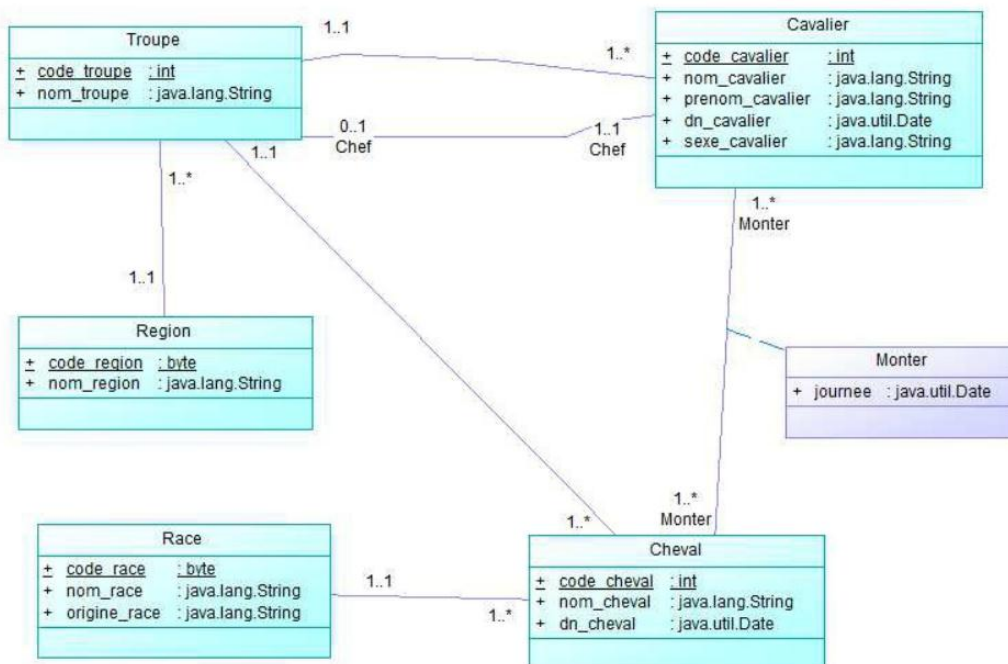
### **a. Diagramme des cas d'utilisation**



### **b. Diagramme de séquence**



c. Diagramme de classes



### **Dossier 3 : BASE DE DONNEES - SQL (10 pts)**

#### **1) Code de création de la base (TRANSACT SQL)**

```

CREATE TABLE [dbo].[EMETTRE] (
    [CODE_CHAINE] [varchar](10) NOT NULL,
    [CODE_SATELLITE] [int] NOT NULL,
    [FREQUENCE] [int] NULL,
    [POLARISATION] [char](1),
    [DEBIT] [int] NULL,
    CONSTRAINT [PK_EMETTRE] PRIMARY KEY CLUSTERED
    (
        [CODE_CHAINE] ASC,
        [CODE_SATELLITE] ASC
    )
)

ALTER TABLE [dbo].[EMETTRE] WITH CHECK ADD CONSTRAINT
[FK_EMETTRE_EMETTRE_CHAINE] FOREIGN KEY ([CODE_CHAINE])
REFERENCES [dbo].[CHAINE] ([CODE_CHAINE])
GO

ALTER TABLE [dbo].[EMETTRE] CHECK CONSTRAINT [FK_EMETTRE_EMETTRE_CHAINE]
GO

ALTER TABLE [dbo].[EMETTRE] WITH CHECK ADD CONSTRAINT
[FK_EMETTRE_EMETTRE2_SATELLIT] FOREIGN KEY ([CODE_SATELLITE])
REFERENCES [dbo].[SATELLITE] ([CODE_SATELLITE])
GO

```

[www.itlearning-settat.com](http://www.itlearning-settat.com)

Email: [admission@itlearning-settat.com](mailto:admission@itlearning-settat.com)

Tél.: 0661077812



```
ALTER TABLE [dbo].[EMETTRE] CHECK CONSTRAINT  
[FK_EMETTRE_EMETTRE2_SATELLIT]
```

**2) Requête permettant d'ajouter la contrainte d'intégrité référentielle reliant la table « Chaine » à la table « Pays »**

Alter table Chaine

ADD FOREIGN KEY (Code-Pays) REFERENCES Pays(Code\_Pays)

**3) Requêtes (TRANSACT SQL)**

```
a)  
select C.*  
from chaine C, emettre E, satellite S  
where C.code_chaine = E.code_chaine  
      and E.code_satellite = S.code_satellite  
      and S.nom_satellite='HOTBIRD'  
  
b)  
select nom_satellite, count(*)  
from satellite S, emettre E  
where S.code_satellite = E.code_satellite  
group by nom_satellite  
  
c)  
select *  
from chaine  
where datediff(day, date_lancement, getdate())<15  
  
d)  
select nom_pays  
from pays P, chaine C  
where P.code_pays = C.code_pays  
group by nom_pays  
having count(*) >= all (select count(*)  
                        from chaine  
                        group by code_pays)
```

**Dossier 4 : Questions (6 pts)**

Voir Guides de Soutien de la filière TDI.

[www.itlearning-settat.com](http://www.itlearning-settat.com)

Email: [admission@itlearning-settat.com](mailto:admission@itlearning-settat.com)

Tél.:0661077812

**Barème de notation :**

**Dossier 1 : (10 points)**

Question	Note
1.	2
2.	2
3.	2
4.	1,5
5.	1,5
6.	1

**Dossier 2 : (14 points)**

Question	Note
a.	5
b.	4
c.	5

**Dossier 3 : (10 points)**

Question	Note
1.	2
2.	2
3.a	1
3.b	1
3.c	2
3.d	2

**Dossier 4 : (6 points)**

Question	Note
4.a	1
4.b	1
4.c	1

[www.itlearning-settat.com](http://www.itlearning-settat.com)

Email: [admission@itlearning-settat.com](mailto:admission@itlearning-settat.com)

Tél.:0661077812

4.d	1
4.e	1
4.f	1

[www.itlearning-settat.com](http://www.itlearning-settat.com)

[www.itlearning-settat.com](http://www.itlearning-settat.com)

Email: [admission@itlearning-settat.com](mailto:admission@itlearning-settat.com)

Tél.:0661077812