



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 passage à la 2<sup>ème</sup> année**

**Session Juin 2016 (Correction)**

**Filière : Techniques de Développement  
Informatique**

**Epreuve : Synthèse**

**Niveau: TS**

**Variante : V2**

**Durée : 5 heures**

**Barème : / 120**

❖ **Partie I : Théorie** (40 pts)

➤ **Dossier 1: Notions de mathématiques appliquées à l'informatique** (12 pts)

1. Convertir en binaire les nombres suivants **(06 pts)**

$(321)_8 : 11010001$

$(C0F)_{16} : 110000001111$

$(45)_{10} : 101101$

2. Effectuer en binaire l'opération suivante **(02 pts)**

$$11111010-11 = 11110111$$

3. A l'aide du tableau de Karnaugh, simplifier la fonction H définie par sa table de vérité suivante : (04 pts)

ef \ g	00	01	11	10
0	1	1	1	1
1	1	0	1	1

Filière	Epreuve	Session	1/11
DI	Synthèse V2(Correction)	Juillet 2016	

$$h = e + f + g$$

➤ **Dossier 2: Techniques de programmation structurée (8 pts)**

```
constante A =12
constante B=10
debut
    i,j,T1[ A],T2[B],res[B][A] : entier ;
    i=0;
    pour i allant de 0 à A
    lire (T1[i]);
    pour i allant de 0 à B
    lire (T2[j]);

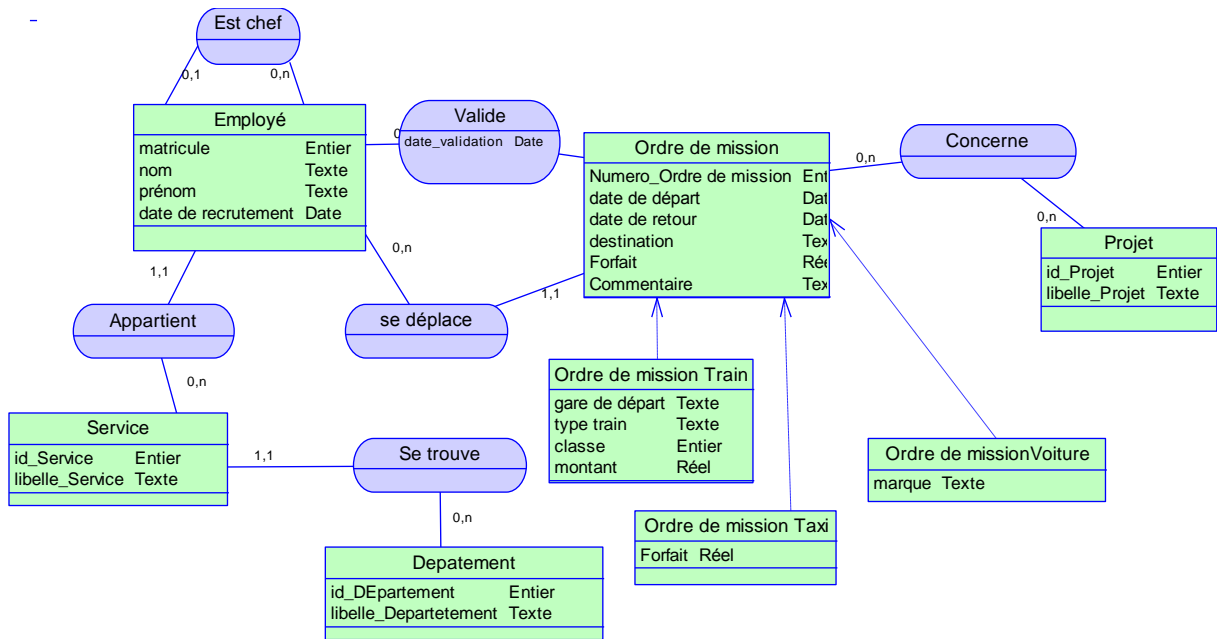
    min= T1[j] ;
    pour i allant de 0 à A
    faire
    if(min < T1[i] );
    min= T1[i]
    fin pour
    pour i allant de 0 à B
    faire
    if(min < T2[i] );
    min= T2[i] ;
    fin pour

    pour i allant de 0 à B
        pour j allant de 0 à A
    res[i][j]=abs (T1[i]*T2[j])*min;
        pour i allant de 0 à B
            pour j allant de 0 à A
    ecrire(res[i][j]);
Fin
```

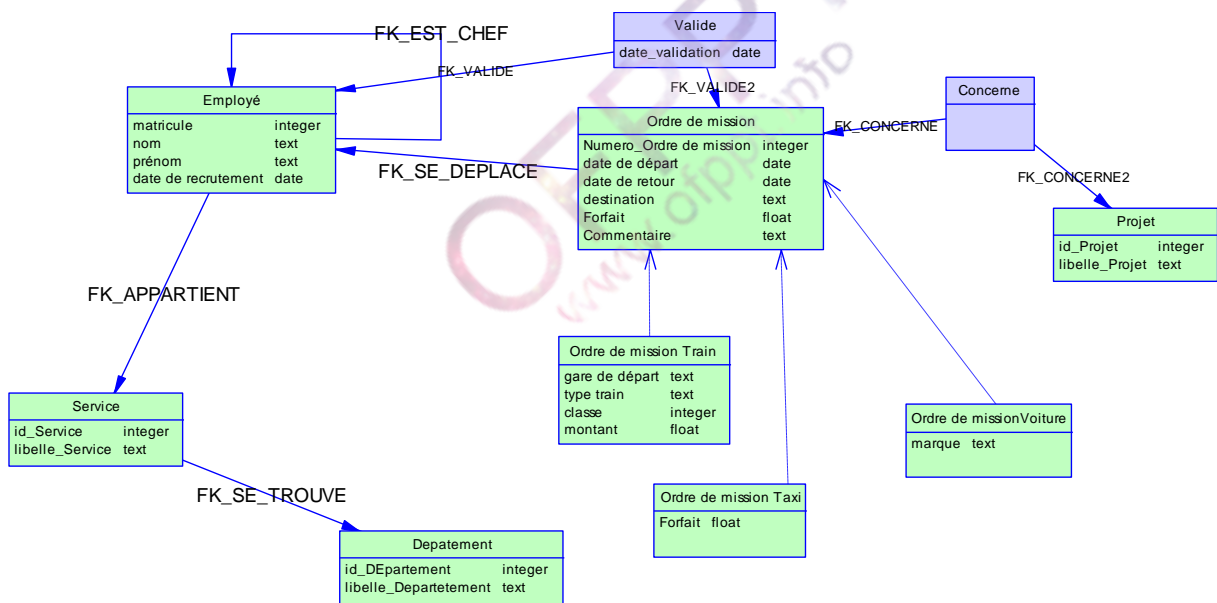
➤ **Dossier 3: Conception et modélisation d'un système d'information (20 pts)**

1. Etablir le modèle conceptuel des données correspondant

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	<b>2/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	



2. Etablir le modèle logique des données associé.



<b>Filière</b>	<b>Epreuve</b>	<b>Session</b>	<b>3/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	

❖ **Partie II: Pratique** (80 pts)

➤ **Dossier 1: Langage de programmation structurée** (25 pts)

1-

```
typedef struct ordreMission {
    int identifiant;
    char mission[30];
    char ville[30];
    int duree;
    char commentaire[30];
    int frais;
    char etat[30];
}nf;
```

2-

***ordreMissionHistorique*** ***OrdreMission*** [30];

3-

```
printf("voulez-vous gerer combien d'ordre de missions ?\n");
scanf("%d", &n);
for(j=0; j<n; j++){
    printf("info de l'ordreMission %d\n", j+1);
    printf("donner l'identifiant de l'ordre de missions \n");
    scanf("%d", &HistoriqueOrdreMission[j].identifiant);
    printf("donner la mission\n");
    scanf("%s", HistoriqueOrdreMission[j].mission);
    printf("donner la ville\n");
    scanf("%s", HistoriqueOrdreMission[j].ville);
    printf("donner la duree de l'ordre de missions \n");
    scanf("%d", &HistoriqueOrdreMission[j].duree);
    printf("donner le frais de l'ordre de missions \n");
    scanf("%d", &HistoriqueOrdreMission[j].frais);
    printf("donner l'etat de ordreMission \n");
    scanf("%s", HistoriqueOrdreMission[j].etat);
}
```

4-

<b>Filière</b>	<b>Epreuve</b>	<b>Session</b>	<b>4/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	

```

for(j=0;j<n;j++){
printf("info de l'ordreMission %d\n",j+1);
printf("le nom %s ",HistoriqueOrdreMission[j].mission);
printf("la ville %s", HistoriqueOrdreMission[j].ville);
printf("l'identifiant de l'ordreMission %d", HistoriqueOrdreMission
[j].identifiant);
printf("la durée %d", HistoriqueOrdreMission[j].duree);
printf("les frais %d", HistoriqueOrdreMission[j].frais);
printf("le type de ordreMission %s", HistoriqueOrdreMission[j].etat);
}

```

5-

```

for(j=0;j<n;j++){
if(HistoriqueOrdreMission[j].etat == " Refusé ")
{
printf("info du ordreMission %d\n",j+1);
printf("le nom %s ",HistoriqueOrdreMission[j].mission);
printf("la ville %s", HistoriqueOrdreMission[j].ville);
printf("l'identifiant de ordreMission %d", HistoriqueOrdreMission
[j].identifiant);
printf("la duree %d", HistoriqueOrdreMission[j].duree);
printf("le montant %d", HistoriqueOrdreMission[j].frais);
printf("L'etat de l'ordreMission %s", HistoriqueOrdreMission[j].etat);
}
}

```

6-

```

inttotfrais=0 ;
for(j=0;j<n;j++){
totfrais+= HistoriqueOrdreMission[j].frais;
printf(" le frais totale des ordreMission est: %d\n", totfrais);
}

```

7-

```

printf("donner le numero et le frais de l'ordre de missions \n");
scanf("%d%d",&num,&mnt);
for(j=0;j<n;j++){
if((HistoriqueOrdreMission [j].mission,identifiant)==num)
HistoriqueOrdreMission [j].frais=mnt;
}

```

8-

```

printf("donner l'identifiant de l'ordre de missions \n");
scanf("%d",&HistoriqueOrdreMission[j].identifiant);
printf("donner la mission\n");
scanf("%s",HistoriqueOrdreMission[j].mission);
printf("donner la ville\n");
scanf("%s",HistoriqueOrdreMission[j].ville);
printf("donner la duree de l'ordre de missions \n");
scanf("%d",&HistoriqueOrdreMission[j].duree);
printf("donner le frais de l'ordre de missions \n");
scanf("%d",&HistoriqueOrdreMission[j].frais);

```

Filière	Epreuve	Session	5/11
DI	Synthèse V2(Correction)	Juillet 2016	

```

printf("donner l'etat de ordreMission \n");
scanf("%s",HistoriqueOrdreMission[j].etat);
i=0;
while(HistoriqueOrdreMission[i].identifiant<ident) i++;
for(j=n-1;j>=i;j--){
HistoriqueOrdreMission [j+1]= HistoriqueOrdreMission [j];
}
strcpy(HistoriqueOrdreMission [i].mission,mission);
strcpy(HistoriqueOrdreMission [i].ville,ville);
HistoriqueOrdreMission [i].identifiant=identifiant;
printf("le nouveau tableau des notes de frais est:\n");
n++;
for(j=0;j<n;j++){
printf("info du ordreMission  %d\n",j+1);
printf("le nom  "%s  ",HistoriqueOrdreMission[j].mission);
printf("la ville  "%s", HistoriqueOrdreMission[j].ville);
printf("le identifiant de ordreMission  %d", HistoriqueOrdreMission
[j].identifiant);
printf("la duree  %d", HistoriqueOrdreMission[j].duree);
printf("le montan  %d", HistoriqueOrdreMission[j].frais);
printf("le type de ordreMission  %s", HistoriqueOrdreMission[j].etat);
}

```

9-

```

do{
printf("*****Menu*****:\n");
printf("*****QUESTION 1*****:\n");
printf("*****QUESTION 2*****:\n");
printf("*****QUESTION 3*****:\n");
printf("*****QUESTION 4*****:\n");
printf("*****QUESTION 5*****:\n");
printf("*****QUESTION 6*****:\n");
printf("*****QUESTION 7*****:\n");
printf("*****QUESTION 8*****:\n");
printf("*****QUESTION 9*****:\n");
printf("*****QUITTER 0*****:\n");
scanf("%d",&choix);
switch(choix){
case 1:
case 2:
etc...
}while(choix!=0);

```

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	6/11
<i>DI</i>	<i>Synthèse V2(Correction)</i>	<i>Juillet 2016</i>	

## Par Employé

### ➤ **Dossier 2: Langage de programmation Orientée Objet (30 pts)**

#### 1. Classe **Employé**. (4 pts)

```
class Employé
{
    int Matricule { get; set; }
    public string Nom { get; set; }
    public string Prenom { get; set; }
    public string Adresse { get; set; }
    public string Genre { get; set; }
    public float Age { get; set; }
    private string service { get; set; }
    private string departement { get; set; }

    public string Ville { get; set; }
    public Employé() { }
    public Employé(int id, string nom, string prenom, string adresse, string
Genre, float Age, string service, string dep)
    {
        this.Matricule = id;
        this.Nom = nom;
        this.Prenom = prenom;
this.Adresse = adresse;
        this.Genre = Genre;
this.Age = Age;
        this.service = service;
        this.departement = dep;
    }

    public override string ToString()
    {
        return "id:"+ this.Matricule + " Nom\n" + this.Nom + "Prenom\n" +
this.Prenom+"Adresse:"+ this.Adresse + " Genre\n" + this.Genre + "Age\n" +this.Age+
"service:"+ this.service + " Departement\n" + this.departement ;
    }
}}
```

#### 2. Classe **Mission**: (4 pts)

```
class Mission
{
    int Numero;
    string Libellé;
    string Lieu;
    string Commentaire;
    float Montant;
    public Mission() { }
    public Mission(int num, string libelle, string lieu, string commentaire,
float montant)
    {
```

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	<b>7/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	

```

        this.Numero = num;
        this.Libellé = libelle;
        this.Lieu = lieu;
        this.Commentaire = commentaire ;
        this.Montant = montant ;

    }

public virtual int CalculerCharge ()
    {
        return montant * taux ;}
    public override string ToString()
    {
        return "Numero:" + this.Numero + " Libellé\n" + this.Libellé + "Lieu\n"
+ this.Lieu + "Commentaire:" + this.Commentaire + " Montant\n" + this.Montant;
    }}

```

### 3. Classe *montantException*: (7 pts)

```

class cheveauException : Exception
    {
        public cheveauException()
            : base("le montant ne peut pas etre inférieur à 10 ou supérieur à 500
")
        {
        }
    }
class OrdreMissionTrain: Mission
    {
        Employe emp;
        int classe;
        string typetrain;
        string montant;
    public OrdreMissionTrain():base() { }
    public OrdreMissionTrain(int num, string libelle, string lieu, string commentaire,
float montant , string emp, int classe, string typetrain,string montant): base
( num, libelle, lieu, commentaire, montant )
    {
        if ((montant < 10) || (classe > 500))
        { throw new cheveauException (); }
        else
        {
            this.emp = emp ;
            this.classe = classe ;
            this.typetrain = typetrain ;
            this.montant = montant ;
        }
    }

    public override int CalculerCharge ()
    {
        return montant * classe ;
    }

    public override string ToString()
    {
        return(base.ToString() + "emp:" + this.emp + " classe\n" + this.classe +
"typetrain\n" + this.typetrain + "montant:" + this.montant);
    }
}

```

Filière	Epreuve	Session	8/11
DI	Synthèse V2(Correction)	Juillet 2016	



#### 4. Classe **ListeMissions**: (10 pts)

```
class ListeMissions
{
    public List<Mission> lp;

    public ListeMissions()
    {
        lp = new List<Mission>();
    }

    public void ajouter(Mission dep)
    {
        Console.WriteLine("Confirmer l'ajout en tapant le chiffre 1");
        int rep=Convert.ToInt32( Console.ReadLine());
        if (rep==1)
            lp.Add(dep);
    }
    public void Afficher()
    {
        foreach (Mission dep in lp)
            dep.ToString();
    }

    public void supprimer(Mission dep)
    {
        foreach (Mission d in lp)
        {
            if (d.Equals(dep))
            {
                Console.WriteLine("Confirmer la suppression en tapant le
chiffre 1");
                int rep = Convert.ToInt32(Console.ReadLine());
                if (rep == 1)
                    lp.Remove(dep); break;
            }
        }
    }

    public void Rechercher()
    {
        foreach (Mission d in lp)
        {
            if (d.Montan > 1000 )
            {
                d.ToString();
            }
        }
    }
}
```

<b>Filière</b>	<b>Epreuve</b>	<b>Session</b>	<b>9/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	

## **Dossier 2 : (25 Pts)**

1. Ecrire le code du bouton **Enregistrer** permettant d'enregistrer la liste des ordres de missions dans un fichier texte. **(6 pts)**

```
private void boutonEnregistrer_Click(object sender, EventArgs e)
{
    FileStream fichier = File.Open("fich.xml", FileMode.Create);
    XmlSerializer s = new XmlSerializer(typeof(ListeMissions));
    s.Serialize(fichier, this);
    fichier.Close();
}
```

2. Ecrire le code du bouton **Afficher** permettant d'afficher les ordres de missions d'un Lieu sélectionnée à partir d'un combobox. **(6 pts)**

```
public List<Mission> lp;

private void boutonAfficher_Click(object sender, EventArgs e)
{
    foreach (Mission d in lp)
    {
        if (d.Lieu1 == comboBox1.Text)
        {
            dataGridView1.Rows.Add(d.Numero1,d.Montan,d.Lieu1, d.Libellé1 ,d.Commentaire1);
        }
    }
}
```

3. Ecrire le code nécessaire pour le bouton **Supprimer** qui permet de supprimer l'ordre de mission dont le numéro est saisi dans le textbox, la suppression doit être effectuée à la fois dans la liste et dans la grille et un message de confirmation doit être affiché avant de procéder à la suppression. **(7 pts)**

```
private void ButtonSupprimer_Click(object sender, EventArgs e)
{
    bool t = false;
    for (int i = 0; i < lp.Count; i++)
    {
        if (lp[i].Numero1 == Convert.ToInt32 ( textBox1.Text))
        {
            t = true;
            DialogResult res = MessageBox.Show("Etes vous sur de vouloir effectuer la suppression?", "Confirmation", MessageBoxButtons.YesNo);
            if (res == DialogResult.Yes)
            {
                for (int j = 0; j < dataGridView1.RowCount; j++)
                {
                    if (Convert.ToInt32 (
                    dataGridView1.Rows[j].Cells[0].Value) == Convert.ToInt32 ( textBox1.Text))
                        dataGridView1.Rows.RemoveAt (j);
                }
            }
        }
    }
}
```

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	<b>10/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	

```

        //dataGridView1.Rows.RemoveAt(i);
        listBox1.Items.Remove(lp[i]);
        lp.RemoveAt(i);
        clear(); break;
    }
}
    }
    if (t == false) MessageBox.Show("ordre de mission inexistant");
}

```

4. Ecrire le code nécessaire pour le bouton **Total** permettant d'afficher le montant total des ordres de missions enregistrés. **(6 pts)**

```

private void buttonTotal_Click(object sender, EventArgs e)
{
    float total = 0.0;

    foreach (Mission d in lp)
    {
        total += d.Montan;
    }

    TextBoxTotal = total.ToString();
}

```

OFPPT  
www.ofppt.info

<i>Filière</i>	<i>Epreuve</i>	<i>Session</i>	<b>11/11</b>
<b>DI</b>	<b>Synthèse V2(Correction)</b>	<b>Juillet 2016</b>	