

Direction Recherche et Ingénierie de la Formation

Examen de passage à la 2^{eme} année

Session Juillet 2014

Filière : Techniques de Développement Informatique

Epreuve : Synthèse

Niveau: TS

Variante : V1

Durée : 5 heures

Barème : / 120 Pts

❖ Partie I : Théorie (40 pts)

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

▪ **Exercice 1: (6 pts)**

NB : la calculatrice est strictement interdite.

Compléter le tableau suivant:

| Décimal | Binaire | Octal | Hexadécimal |
|---------|----------|-------|-------------|
| 1796 | | | |
| | 10101100 | | |
| | | 1767 | |
| | | | E10 |

▪ **Exercice 2: (6 pts)**

Soit la fonction logique suivante:

$$Q(k, l, m, n) = \overline{m}\overline{l}n + \overline{m}\overline{k}\overline{l} + lm\overline{n}$$

- 1- Simplifier analytiquement la fonction logique Q . (2 pts)
- 2- Construire la table de vérité. (2 pts)
- 3- Simplifier avec la méthode de Karnaugh la fonction logique Q . (2 pts)

➤ **Dossier 2: Techniques de programmation structurée (16 pts)**

▪ **Exercice 1: Vérification d'un tableau trié (8 pts)**

Ecrire une fonction "`est_trié()`" permettant de vérifier si un tableau de réelles T de taille N (passés en argument de cette fonction) est trié ou non. La fonction doit renvoyer l'un des trois messages: « *trié croissant* », « *trié décroissant* » ou « *non trié* ». Faites appel à cette fonction dans un algorithme donnant le droit à l'utilisateur de remplir un tableau avec N valeurs.

| Filière | Epreuve | Session | 1/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V1 | Juillet 2014 | |

Exemple:

Donner le nombre des éléments du tableau:

4

Donner les éléments du tableau:

-15.33

-8.54

-3.68

19.75

Votre tableau est trié croissant.

■ **Exercice 2: Le cerf-volant (8 pts)**

Ecrire une procédure "*cerf_volant()*" permettant de dessiner un cerf-volant dont la taille N est passée en paramètre de la procédure. Faites appel à cette procédure dans un algorithme donnant le droit à l'utilisateur de saisir la taille du dessin qui doit être paire.

Exemple:

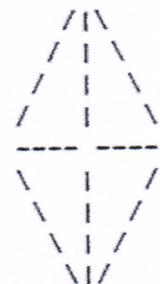
Donner la taille paire:

3

La taille doit être paire!

Donner la taille paire:

8



Dossier 3: Conception et modélisation d'un système d'information (12 pts)

Conception d'un système d'information pour une société informatique

Une grande société de fabrication de produits informatiques souhaite réaliser un site web, permettant de présenter ses produits à la clientèle. La société a plusieurs filiales installées dans plus d'une cinquantaine de pays à travers le monde, réparties sur 8 zones géographiques: Amérique du nord, Amérique latine, Europe, Afrique, Moyen-Orient, Asie, Extrême-Orient et Australie. Dans sa politique commerciale, la société propose pour chaque pays un ensemble de produits s'adaptant au pouvoir d'achat des citoyens du pays.

L'interface du site web permet de rediriger automatiquement un internaute sur la page de son pays pour avoir accès à la liste des produits disponibles dans son pays. Les produits quant à eux sont répertoriés sous plusieurs catégories (par exemple: Ordinateurs, Imprimantes,...). On choisissant la catégorie "Ordinateurs" par exemple, l'intéressé sera appelé à choisir entre "Postes de travail", "Ordinateurs portables",.... Par contre, il y a des catégories comme "Scanners" où l'utilisateur se voit directement devant la liste de produits sans avoir à passer par un deuxième choix.

La société donne aux clients de certains pays le droit d'effectuer des achats directement à partir du site. Pour effectuer un achat, le client doit s'enregistrer dans le site en fournissant son nom et prénom, son adresse, sa ville (à partir d'une liste déroulante), son code postal (généré automatiquement après sélection

| Filière | Epreuve | Session | 2/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse VI | Juillet 2014 | |

de la ville), son numéro de téléphone et son e-mail. L'enregistrement ne se valide sauf si le client achète au moins un produit, en spécifiant la quantité achetée du produit.

- 1- Etablir le dictionnaire de données. (3 pts)
- 2- Etablir le modèle conceptuel de données. (6 pts)
- 3- Etablir le modèle logique de données. (3 pts)

❖ Partie II: Pratique (80 pts)

➤ Dossier 1: Langage de programmation structurée (20 pts)

▪ Exercice 1: (10 pts)

Soit le programme C suivant :

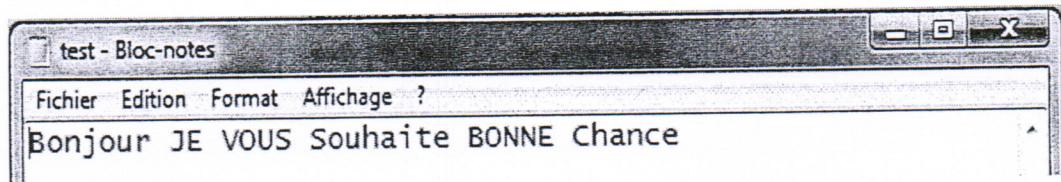
```
#include <stdio.h>
#include <ctype.h>
int main()
{
    FILE *fpi=fopen("c:\\ofppt\\test.txt","r");
    FILE *fpo=fopen("c:\\ofppt\\res.txt", "w");
    int c;
    for(;(c=fgetc(fpi)) != EOF;){
        if(c == ' ') // test sur le caractère espace
            fputc('\\n', fpo);
        else
            fputc(((toupper(c) + 5 - 'A') % 26) + 'A' , fpo);

    }
    getchar();
}
```

NB :

- i. *toupper* c'est une fonction de la bibliothèque *ctype.h* qui convertit un caractère en majuscule.
- ii. Les codes ASCII des caractères de 'A' à 'Z' sont ordonnés. Si le code de 'A' est une valeur α alors celui de 'B' est $\alpha+1$, ... celui de 'Z' est $\alpha+25$.

Le fichier « test.txt » contient le texte suivant :



- a. Que fournit l'expression : « ((toupper(c) + 5 - 'A')%26) + 'A' » pour les valeurs suivantes de **c** : 'B', 'L', 'V' et 'Z'. C'est quoi donc le rôle de cette expression. (4 pts)
- b. Quel est le contenu de fichier « res.txt » après exécution du programme. (6 pts)

| Filière | Epreuve | Session | 3/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse VI | Juillet 2014 | |

- **Exercice 2 : (10 pts) :** Quel est le résultat d'affichage du programme suivant

```
#include <stdio.h>
int gct(int *k)
{
    int x=6;
    printf("Debut gct : x=%d, *k=%d\n", x, *k); //2 pts
    x=*k+1;
    printf("fin gct: x=%d, *k=%d\n", x, *k); // 2 pts
    return(x * *k);

}
void fct(int n, int *m)
{
    printf("Debut fct n=%d, *m=%d\n", n, *m); //1 pt
    n=*m+3;
    *m=gct(&n);
    printf("Fin fct n=%d, *m=%d\n", n, *m); //2 pts
}
int main()
{
    int a=5, b=3;
    printf("Debut main a=%d, b=%d\n", a, b); //1 pt
    fct(a, &b);
    printf("Fin main a=%d, b=%d\n", a, b); //2 pts
    getchar();
}
```

➤ Dossier 2: Programmation orientée objet (35 pts)

Développement d'une application orientée objet pour la société informatique

On souhaite développer une application orientée objet à l'instar du système conçu pour la société de fabrication des produits informatiques. L'application contiendra plusieurs classes dont tous les attributs doivent être **privés**, ce qui implique le besoin de coder des **accesseurs** dans toutes les classes. Les accesseurs des attributs affectées automatiquement doivent être en **lecture seule**. Chaque classe doit avoir un **constructeur d'initialisation**, et une méthode "ToString()".

On suppose que les classes suivantes sont déjà créées :

- **Classe "Zone"** : Ayant comme attributs un identifiant pour la zone géographique (entier positif qui s'incrémente automatiquement), et le nom de la zone géographique

```
public class Zone
{
    private uint IdZone;
    private string NomZone;
    private static uint auto;
}
```

- **Classe "Pays"** : Ayant comme attributs un identifiant pour le pays (entier positif qui s'incrémente automatiquement), le nom du pays, et sa zone géographique.

```
public class Pays
{
    private uint IdPays;
    private string NomPays;
    private Zone zone;
    private static uint auto;}
```

| Filière | Epreuve | Session | 4/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V1 | Juillet 2014 | |

- **Classe "Ville"** : Ayant comme attributs un identifiant pour la ville (entier positif qui s'incrémente automatiquement), le nom de la ville, son code postal (entier positif) et le pays où se situe la ville.

```
public class Ville
{
    private uint IdVille;
    private string NomVille;
    private uint CodePostal;
    private Pays pays;
    private static uint auto;}
```

- **Classe "Catégorie"** : Ayant comme attributs un identifiant pour la catégorie (entier positif qui s'incrémente automatiquement), le nom de la catégorie, et la catégorie parente, et un 2^{ème} constructeur avec un seul paramètre (nom de la catégorie).

```
public class Catégorie
{
    private uint IdCatégorie;
    private string NomCatégorie;
    private Catégorie catégorie;
    private static uint auto;}
```

- **Classe "Produit"** : Ayant comme attributs un identifiant du produit (entier positif qui s'incrémente automatiquement), le nom du produit, son prix et sa catégorie.

```
public class Produit
{
    private uint IdProduit;
    private string NomProduit;
    private float PrixProduit;
    private Catégorie catégorie;
    private static uint auto;}
```

Vous devez répondre aux questions suivantes :

- 1- Dans la classe "Pays", ajouter un nouvel attribut "Produits_disponibles" qui sera une liste générique des produits disponibles dans le pays. (2,5 pts)
- 2- Dans la classe "Pays", ajouter une méthode "AjouterProduit(produit)" permettant d'ajouter un nouveau produit à la liste des produits disponibles dans le pays. Une exception est levée si le produit est déjà existant (produit ayant le même identifiant). (3,5 pts)
- 3- Dans la classe "Pays", ajouter une méthode "SupprimerProduit(identifiant_produit)" permettant de supprimer un produit en fournissant son identifiant. Une exception est levée si l'identifiant du produit est inexistant. (3,5 pts)
- 4- Dans la classe "Pays", ajouter une méthode "string ListerProduits()" permettant de lister les produits disponibles dans le pays. (2,5 pts)

- **Classe "Achat"**

- 5- Créer une classe "Achat" ayant comme attributs le produit acheté, la quantité achetée, et la date d'achat (affectée automatiquement par la date système). (3,5 pts)
- 6- Ajouter une méthode "MontantAchat()" permettant de calculer le montant d'un achat. (2,5 pts)

| Filière | Epreuve | Session | 5/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse VI | Juillet 2014 | |

▪ Classe "Client"

- 7- Créer une classe "Client" ayant comme attributs un identifiant du client (entier positif qui s'incrémente automatiquement), son nom et son prénom, son adresse postale, sa ville (de type "ville") son numéro de téléphone, son adresse e-mail et la liste des achats qu'il a effectuées. (4 pts)
- 8- Ajouter une méthode "AcheterProduit(achat)" permettant d'ajouter un achat à la liste des achats du client. (3 pts)
- 9- Ajouter une méthode "string ListerAchats(date_début, date_fin)" permettant de lister les achats effectués par le client entre une date de début et une date de fin. (4 pts)

▪ Classe "Client Fidèle"

Un client fidèle "**est un client**" qui s'adhère au système de fidélisation de la société en payant un montant annuel pour l'abonnement, ce qui lui permet de bénéficier d'un numéro de fidélité et avoir une réduction de 25% sur ses achats.

- 10- Créer la classe "Client_Fidèle" ayant comme attributs un numéro de fidélité (entier positif qui s'incrémente automatiquement). (3 pts)
- 11- Redéfinir la méthode "AcheterProduit(achat)" permettant d'ajouter un achat à la liste des achats du client après avoir appliqué une réduction de 25% sur le prix du produit acheté. (3 pts)

➤ Dossier 3: Programmation événementielle (25 pts)

NB : Dans ce dossier, on vous demande de donner uniquement le code à mettre à l'intérieur des méthodes événementielles. L'entête de ces méthodes événementielles n'est pas demandé !

Soit le formulaire suivant nommée "Liste des achats":

| | Nom Produit | Quantité_Achetée | Date Achat | Montant Achat |
|----|-------------|------------------|---------------------|---------------|
| 1 | NetBook | 2 | 20/01/2014 15:46:23 | 5980 DH |
| 2 | NoteBook | 2 | 20/01/2014 15:46:23 | 9980 DH |
| 3* | | | | |

- 1- Dans le code du formulaire on a déclaré un objet de type "Client":

```
Client c1;
```

Dans la méthode de chargement du formulaire on a instancié l'objet "Client":

```
c1 = new Client("ALAMI", "Ahmed", "25, Rue Riad, Qu. Oummnia", rabat, "0689741211", "alami@mail.ma");
```

| Filière | Epreuve | Session | 6/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse VI | Juillet 2014 | |

Ajouter dans la méthode de chargement du formulaire, le code permettant d'afficher l'en-tête du formulaire comme ce qui est indiqué dans l'image ci-dessus. (5 pts)

- 2- Ecrire le code du bouton "Afficher" permettant d'afficher la liste des achats du client "c1", effectuées entre une date de début et une date de fin, sélectionnées par l'utilisateur (voir image ci-dessus). (8 pts)
- 3- Ecrire le code du bouton "Sauvegarder" qui permet d'enregistrer la liste des achats du client "c1", effectuées entre une date de début et une date de fin, sélectionnées par l'utilisateur (voir image ci-dessus) dans un fichier *objet* (sérialisation) qui porte le nom « AchatsAlami.dat ». (8 pts)
- 4- Ecrire le code du bouton "Effacer" permettant d'effacer le contenu du composant graphique représentant la liste, et d'initialiser les sélecteurs des dates à la date système. (4 pts)

| Filière | Epreuve | Session | 7/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V1 | Juillet 2014 | |



Direction Recherche et Ingénierie de la Formation

Examen de passage à la 2^{ème} année

Session Juillet 2014

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)**

Exercice 1: (6 pts)

NB : la calculatrice est strictement interdite.

Compléter le tableau suivant:

| Décimal | Binaire | Octal | Hexadécimal |
|---------|-----------|-------|-------------|
| 6548 | | | |
| | 110011011 | | |
| | | 5643 | |
| | | | 23BC |

Exercice 2: Algèbre de Boole (6 pts)

Soit la table de vérité de la fonction logique \bar{Q} :

| k | I | m | n | Q |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

- 1- Donner la 1^{ère} forme canonique de la fonction logique Q . (1,5 pts)
- 2- Simplifier analytiquement la fonction logique Q . (2 pts)
- 3- Simplifier la fonction logique Q avec la méthode de Karnaugh. (2,5 pts)

| <i>Filière</i> | <i>Epreuve</i> | <i>Session</i> | 1/7 |
|----------------|----------------|----------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

➤ **Dossier 2: Techniques de programmation structurée (16 pts)**

Exercice 1: Le sablier (8 pts)

Ecrire une procédure "*sablier()*" permettant de dessiner un sablier dont la taille N est passée en paramètre de la fonction. Faites appel à cette fonction dans un algorithme donnant le droit à l'utilisateur de saisir la taille du dessin qui doit être paire.

Exemple:

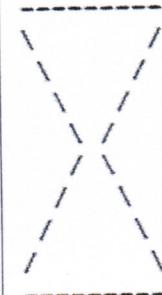
Donner la taille paire:

5

La taille doit être paire!

Donner la taille paire:

8



Exercice 2: Elimination des répétitions (8 pts)

Ecrire une fonction "*sans_répétition()*" permettant de supprimer les répétitions des lettres dans un tableau **T** de taille **N** (passés en argument de cette fonction). La fonction doit renvoyer le mot constitué des lettres restantes. Faites appel à cette fonction dans un algorithme donnant le droit à l'utilisateur de remplir un tableau avec **M** lettres.

Exemple:

Donner le nombre des lettres:

6

Donner les lettres:

E

E

S

T

E

S

Votre mot est EST.

| Filière | Epreuve | Session | 2/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

Dossier 3: Conception et modélisation d'un système d'information (12 pts)

Conception d'un système d'information pour une société informatique

Une grande société de fabrication de produits informatiques souhaite réaliser un site web, permettant de présenter ses produits à la clientèle. La société fabrique plusieurs produits informatiques, notamment des serveurs, des postes de travail, des ordinateurs portables, des imprimantes,

Certains produits de la société informatique nécessitent l'intégration de composants de constructeurs tierces. Par exemple: pour fabriquer un ordinateur personnel, la société peut utiliser les microprocesseurs et les mémoires vives fabriquées par des constructeurs spécialisés en microélectronique. On indique par conséquent le nombre de composants utilisés (Par exemple: 3 barrettes mémoire, 2 disques dur,...). D'autre part, certains produits (ordinateurs, tablettes, ...) ont besoin de systèmes d'exploitation pour les utiliser; ces systèmes sont développés par des sociétés de développement informatique. La société n'installe qu'un système d'exploitation par machine.

Pour construire ses produits informatiques, la société informatique peut faire appel à des constructeurs de composants informatiques. Par exemple: pour fabriquer un ordinateur personnel, la société peut introduire des disques dur fabriquées par une société spécialisée dans ce domaine. Les constructeurs peuvent être des filiales d'autres sociétés constructrices en conséquence des fusions qui se font entre les sociétés. Par exemple: pour ses ordinateurs, la société informatique peut utiliser les microprocesseurs d'un constructeur, et les cartes graphiques d'une filiale de ce même constructeur.

Pour faire fonctionner les composants de ses produits, la société informatique installe les pilotes de chaque composant tiers, en tenant compte de la compatibilité avec le système d'exploitation utilisé avec le produit.

Pour chaque constructeur ou société de développement informatique, on retient sa raison sociale, son adresse postale, son numéro de téléphone, son e-mail et le lien vers son site web. Pour chaque composant, produit, pilote ou système d'exploitation, on retient sa référence, sa désignation et son prix (sauf pour les pilotes qui sont fournies gratuitement avec leurs composants).

1. Etablir le dictionnaire de données. (3 pts)
2. Etablir le modèle conceptuel de données. (6 pts)
3. Etablir le modèle logique de données. (3 pts)

❖ Partie II : Pratique (80 pts)

➤ Dossier 1: Langage de programmation structurée (20 pts)

Exercice 1: Soit le programme C suivant :

```
#include      <stdio.h>
#include      <ctype.h>
int main()
{
    FILE      *fpi=fopen("c:\\ofppt\\test.txt","r");
    FILE      *fpo=fopen("c:\\ofppt\\res.txt", "w");
    int c;
    for(;(c=fgetc(fpi)) != EOF;){
        if(c == ' ') // test sur le caractère espace
            fputc('$', fpo);
        else
            fputc(((toupper(c)- 5 + 'A') % 26 ) + 'A' , fpo);

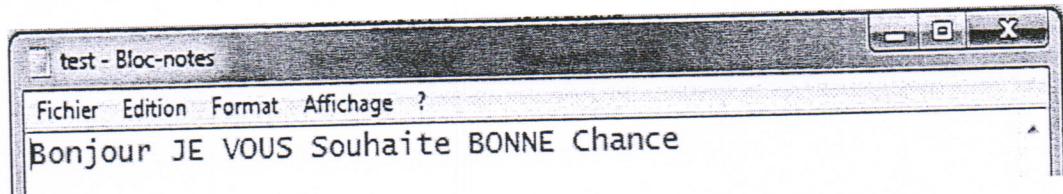
    }getchar();}
```

| Filière | Epreuve | Session | 3/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

NB :

- i. *toupper* c'est une fonction de la bibliothèque *ctype.h* qui convertit un caractère en majuscule.
- ii. Les codes ASCII des caractères de 'A' à 'Z' sont ordonnés. Si le code de 'A' est une valeur α alors celui de 'B' est $\alpha + 1$, ... celui de 'Z' est $\alpha + 25$.

Le fichier « test.txt » contient le texte suivant :



- a. Que fournit l'expression : « ((toupper(c) - 5 + 'A')%26) + 'A' » pour les valeurs suivantes de **c** : 'M', 'H', 'D' et 'A'. Déduire son rôle. (4 pts)
- b. Quel est le contenu de fichier « res.txt » après exécution de programme. (6 pts)

Exercice 2 : (10 pts)

Quel est le résultat d'affichage du programme suivant :

```
#include <stdio.h>
int chose(int *k)
{
    int x=3;
    printf("Debut chose : x=%d, *k=%d\n", x,*k); // 2 pts
    *k = *k + 1; x=*k;
    printf("fin chose: x=%d, *k=%d\n",x, *k); // 2 pts
    return(x**k);

}
void test(int n, int *m)
{
    printf("Debut test n=%d, *m=%d\n", n, *m); // 1 pt
    n=*m*3;
    *m=chose(&n);
    printf("Fin test n=%d, *m=%d\n", n, *m); // 2 pts
}
int main()
{
    int a=1, b=2;
    printf("Debut main a=%d, b=%d\n", a, b); // 1 pt
    test(a,&b);
    printf("Fin main a=%d, b=%d\n", a, b); // 2 pts
    getchar();
}
```

➤ Dossier 2: Programmation orientée objet (35 pts)

Développement d'une application orientée objet pour la société informatique

On souhaite développer une application orientée objet à l'instar du système conçu pour la société de fabrication des produits informatiques. L'application contiendra plusieurs classes dont tous les attributs doivent être privés (ou protégés), ce qui implique le besoin de coder des accesseurs dans toutes les

| Filière | Epreuve | Session | 4/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

classes. Les accesseurs des attributs affectées automatiquement doivent être en **lecture seule**. Chaque classe doit avoir un **constructeur d'initialisation**, et une méthode "ToString()".

On suppose que les classes suivantes sont déjà créées :

Classe "Développeur" : Ayant comme attributs un identifiant pour la société de développement (entier positif qui s'incrémente automatiquement), sa raison sociale, son adresse, son numéro de téléphone, son adresse e-mail, et le lien de son site web.

```
public class Développeur
{
    private uint IdDéveloppeur;
    private string RaisonSocialeDéveloppeur;
    private string AdresseDéveloppeur;
    private string TélDéveloppeur;
    private string EMailDéveloppeur;
    private string SiteWebDéveloppeur;
    private static uint auto;
}
```

Classe "Constructeur" : Ayant comme attributs un identifiant pour la société de fabrication des composants (entier positif qui s'incrémente automatiquement), sa raison sociale, son adresse, son numéro de téléphone, son adresse e-mail, et le lien de son site web, et un 2^{ème} constructeur avec un paramètre supplémentaire: **constructeur**, qui exprime la société qui regroupe la filiale en instance.

```
public class Constructeur
{
    private uint IdConstructeur;
    private string RaisonSocialeConstructeur;
    private string AdresseConstructeur;
    private string TélConstructeur;
    private string EMailConstructeur;
    private string SiteWebConstructeur;
    private Constructeur constructeur;

    private static uint auto;
}
```

Classe "Système" : Ayant comme attributs la référence du système d'exploitation, la désignation du système, le prix du système, et le développeur du système.

```
public class Système
{
    private string RéfSystème;
    private string DésignationSystème;
    private float PrixSystème;
    private Développeur dévelopeur;
}
```

Classe "Composant" : Ayant comme attributs la référence du composant, la désignation du composant, le prix du composant, et le constructeur du composant.

```
public class Composant
{
    private string RéfComposant;
    private string DésignationComposant;
    private float PrixComposant;
    private Constructeur constructeur;
}
```

Classe "Intégration" : Ayant comme attributs le composant intégré, et la quantité à intégrer de ce composant dans un produit.

```
public class Intégration
{
    private Composant ComposantIntégré;
    private uint QuantitéIntégrée;
}
```

| Filière | Epreuve | Session | 5/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

Vous devez répondre aux questions suivantes :

Classe "Produit"

- 1- Créer une classe "Produit" ayant comme attributs la référence du produit, la désignation du produit, une liste générique des composants intégrés dans le produit, et le prix du produit. (4 pts)
- 2- Ajouter une méthode "IntégrerComposant(composant_intégré)" permettant d'intégrer un nouveau composant à la liste des composants intégrés dans le produit. Une exception est levée si le composant intégré est déjà existant (composant ayant la même référence). (2,5 pts)
- 3- Ajouter une méthode "EnleverComposant(référence_composant)" permettant d'enlever un composant de la liste des composants intégrés en fournissant sa référence. Une exception est levée si la référence du composant est inexistante. (2,5 pts)
- 4- Ajouter une méthode "CalculerPrix()" permettant de calculer le prix d'un produit obtenu par la somme des montants des composants intégrés (en tenant compte de leurs quantités) plus une majoration de 20%. (4 pts)

Classe "Pilote"

- 5- Créer une classe "Pilote" ayant comme attributs la référence du pilote, la désignation du pilote, la liste générique des composants compatibles, et la liste générique des systèmes d'exploitation compatibles. (4 pts)
- 6- Ajouter une méthode "AjouterComposant(composant)" permettant d'ajouter un nouveau composant à la liste des composants compatibles avec le pilote. Une exception est levée si le composant est déjà existant (composant ayant la même référence). (2,5 pts)
- 7- Ajouter une méthode "AjouterSystème(système)" permettant d'ajouter un nouveau système à la liste des systèmes d'exploitation compatibles avec le pilote. Une exception est levée si la référence du système d'exploitation est déjà existante. (2,5 pts)

Classe "Produit_Pro"

Un produit pro "est un produit" destiné aux professionnels, se caractérisant par sa performance, sa résistance et sa durabilité. Pour de tels produits, la société informatique offre des garanties allant de 2 à 5 ans.

- 8- Créer une classe "Produit_Pro" ayant comme attribut la garantie qui doit être entre 2 et 5 ans, et la date de début de garantie. Une garantie non incluse dans cet intervalle est susceptible de lever une exception dans tout constructeur ou méthode ayant accès à la garantie. (4 pts)
- 9- Redéfinir la méthode "CalculerPrix()" en ajoutant au prix du produit le montant de garantie égale à 750 MAD par année de garantie. (3 pts)
- 10- Ajouter une méthode "Fin_garantie()" qui retourne la date de fin de la garantie d'un produit pro. (3 pts)
- 11- Ajouter une méthode "Etendre_garantie(nombre_années)" permettant d'augmenter le nombre d'années de garantie pour un produit ayant moins de 5 ans de garantie, vers une garantie inférieure ou égale à 5 ans (en tenant compte de l'exception si la garantie totale risque d'excéder 5 ans). (3 pts)

| Filière | Epreuve | Session | 6/7 |
|---------|-------------|--------------|-----|
| TDI | Synthèse V2 | Juillet 2014 | |

➤ Dossier 3: Programmation événementielle (25 pts)

- NB : Dans ce dossier, on vous demande de donner uniquement le code à mettre à l'intérieur des méthodes événementielles. L'entête de ces méthodes événementielles n'est pas demandé !

Soit le formulaire suivant:

| Réf. Composant | Désignation | PU | Quantité Intégrée | Constructeur |
|----------------|---------------------|------|-------------------|--------------|
| R19 | SonTec Premium III | 1000 | 1 | SonTec |
| P305 | SonTec Graphics R-R | 2500 | 1 | SonTec |

- 1- Dans le code du formulaire on a déclaré un objet de type "Produit":

```
Produit powerspeed;
```

Dans la méthode de chargement du formulaire on a instancié l'objet "Produit":

```
powerspeed = new Produit("Y12.1", "PowerSpeed Y12", picos);
```

Ajouter dans la méthode de chargement du formulaire, le code permettant d'afficher la désignation du produit comme titre dans le formulaire (voir image ci-dessus). (3 pts)

- 2- Ajouter dans la méthode de chargement du formulaire, le code permettant d'afficher la liste des composants intégrés dans le produit "powerspeed" (voir image ci-dessus). (6 pts)
- 3- Ecrire le code du bouton "Supprimer" permettant d'enlever un composant du produit en sélectionnant la ligne à supprimer puis en cliquant sur le bouton "Supprimer". Utiliser la méthode "EnleverComposant(référence_composant)" de la classe "Produit" (on ne traite pas le cas si aucune ligne n'est sélectionnée). (8 pts)
- 4- Ecrire le code du bouton "Sauvegarder" qui permet d'enregistrer la liste des composants intégrés dans le produit "powerspeed" (voir image ci-dessus) dans un fichier *objet* (sérialisation) qui porte le nom « ComposantPowerspeed.dat ». (8 pts)

| | | | |
|---------|-------------|--------------|-----|
| Filière | Epreuve | Session | 7/7 |
| TDI | Synthèse V2 | Juillet 2014 | |