

	مكتب التكوين المهني وإنعاش الشغل		
	Office de la Formation Professionnelle et de la Promotion du Travail		
	Direction de la Recherche et de l'Ingénierie de la Formation : Division Examen		

**Examen National de Fin d'année
Session de juin 2019**

Examen de Passage (Epreuve de Synthèse)

Eléments de correction

<u>Filière</u>	<i>Techniques de Développement Informatique</i>	<u>Variante</u>	V2
----------------	---	-----------------	----

<u>Niveau</u>	TS	<u>Durée</u>	4 Heures	<u>Barème</u>	/100
---------------	----	--------------	----------	---------------	------

Consignes et Précisions aux correcteurs :

Chers Mesdames et messieurs les correcteurs, veuillez SVP respecter impérativement les consignes suivantes :

- Tenir compte de la plate-forme utilisée (.Net, Java, ...)
- Plusieurs solutions peuvent être envisagées pour une même question.

1. 0.5 pt pour chaque conversion

6 pts

Binaire	Octal	Décimal	Hexadécimal
10101010	252	170	AA
00110011	63	51	33
00010001	21	17	11
00101100110 1	1315	717	2CD

2.

$$2.1. f(a, b, c, d) = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + ab\bar{c} + abc$$

1.5 pts

$$f(a, b, c, d) = \bar{a}\bar{b}(\bar{c} + c) + \bar{a}b\bar{c} + ab(\bar{c} + c)$$

$$f(a, b, c, d) = \bar{a}\bar{b} + \bar{a}b\bar{c} + ab$$

$$f(a, b, c, d) = \bar{a}(\bar{b} + b\bar{c}) + ab \quad \text{Ou} = \bar{a}\bar{b} + b(\bar{a}\bar{c} + a)$$

1.5 pts

$$f(a, b, c, d) = \bar{a}(\bar{b} + \bar{c}) + ab \quad \text{Ou} = \bar{a}\bar{b} + b(\bar{c} + a)$$

$$f(a, b, c, d) = \bar{a}\bar{b} + \bar{a}\bar{c} + ab \quad \text{Ou} = \bar{a}\bar{b} + b\bar{c} + ab$$

a. Simplification moyennant le tableau de Karnaugh :

	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
$\bar{c}$	1	1	1	0
c	1	0	1	0

$$f(a, b, c, d) = \bar{a}\bar{b} + \bar{a}\bar{c} + ab$$

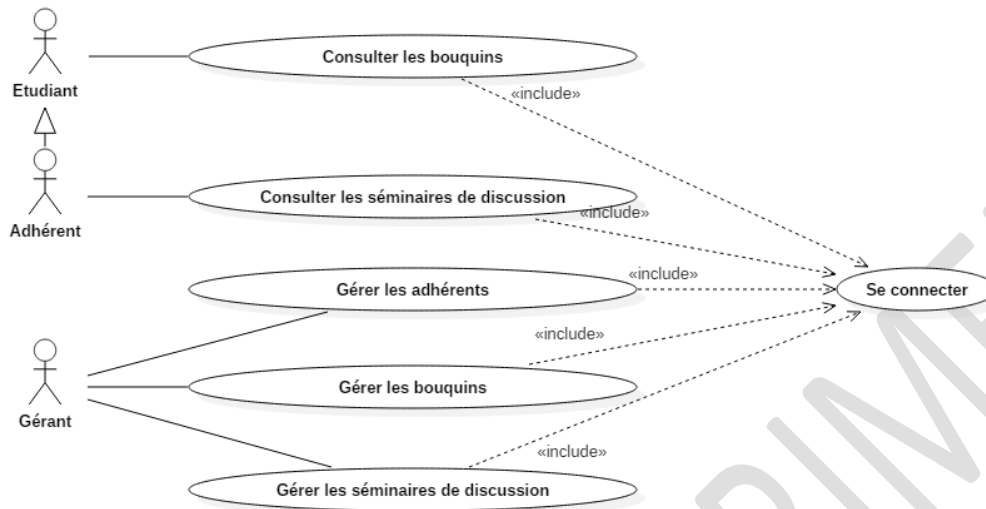
Ou :

$$f(a, b, c, d) = \bar{a}\bar{b} + b\bar{c} + ab$$

3 pts

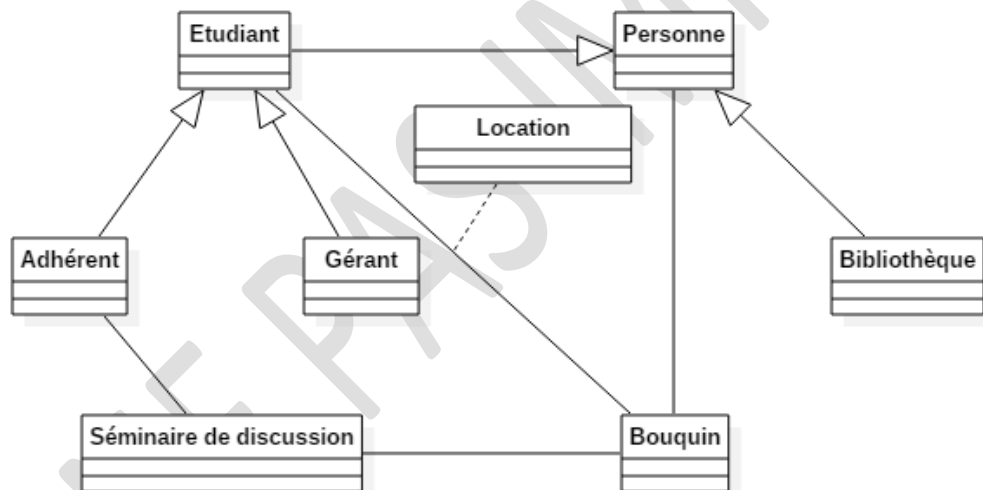
1. Diagramme de cas d'utilisation : 0.25 pt pour chaque élément (acteur, Cas d'utilisation et association)

5 pts



2. Diagramme de classes : 0.5 pt pour chaque élément (classe et association)

8 pts



Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 2 8

Algorithme Ex01

Variables choix : Caractère
flag : Booléen
x, y, somme : Entier

Début

Flag <- FAUX

Répéter

Ecrire("1. Donner x et y\n")
Ecrire("2. Afficher la somme des nombres entre x et y\n")
Ecrire("3. Quitter\n")

Ecrire("\nVotre choix\n")
Lire(choix)

Selon (choix) Faire

1 :

flag <- VRAI

Ecrire("Donner x : ")
Lire(y)

Répéter

Ecrire("Donner y : ")
Lire(y)

TantQue(Non(y > x))

2 :

Si(flag = FAUX)
Ecrire("Vous devez donner x et y")

Sinon

somme <- 1

Pour i <- x à y Faire
somme <- somme + i
FinPour

Ecrire("La somme est : ", somme)

FinSi

3 :

Ecrire("Merci")

Sinon :

Ecrire("Choix invalide !!!")

FinSelon

TantQue(choix != '3')

Fin

Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 3 8

1.

1.1.

3 pts

```

int min(int t[], int n)
{
    int m = t[0], i;

    for(i = 1; i <= n - 1; i++)
    {
        if(t[i] < m)
            m = t[i];
    }

    return m;
}

```

1.2.

2 pts

```

void main()
{
    int t[] = {12, -4, 8, 5, 16, 21, 0, 6};

    printf("%d", min(t, 8));
}

```

2.

7 pts

```

bbbbbb
oooooo
nnnnn
ssss
ooo
ii
r

```

1.

```

[Serializable]
class Etudiant
{
    private string _CodeInscription;
    public string CodeInscription
    {
        get { return _CodeInscription; }
        set
        {
            Match match = Regex.Match(value, @"^\d{12}$");

            if (!match.Success)
                throw new Exception("Code d'inscription invalide !!!");

            _CodeInscription = value;
        }
    }

    private string _NomComplet;
    public string NomComplet
    {
        get { return _NomComplet; }
        set { _NomComplet = value; }
    }

    public Etudiant()
    {
    }

    public Etudiant(string codeInscription, string nomComplet)
    {
        CodeInscription = codeInscription;
        NomComplet = nomComplet;
    }

    public override bool Equals(object obj)
    {
        Etudiant étudiant = obj as Etudiant;

        if (étudiant == null)
            return false;

        return CodeInscription.Equals(étudiant.CodeInscription);
    }

    public override string ToString()
    {
        return string.Format("Etudiant {0} : {1}", CodeInscription, NomComplet);
    }
}

```

2.

```

class Atelier
{
    private List<Etudiant> _Adhérents = new List<Etudiant>();
    public List<Etudiant> Adhérents
    {
        get { return _Adhérents; }
    }
}

```

Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 5 8

```

        set { _Adhérents = value; }
    }

    public Atelier()
    {
    }

    public int NombreAdhérents
    {
        get { return Adhérents.Count; }
    }

    public int IndiceDe(Etudiant adhérent)
    {
        foreach (Etudiant item in Adhérents)
        {
            if (item.Equals(adhérent)) return Adhérents.IndexOf(adhérent);
        }

        return -1;
    }

    public bool Ajouter(Etudiant adhérent)
    {
        if (IndiceDe(adhérent) != -1)
            return false;
        else
        {
            Adhérents.Add(adhérent);
            return true;
        }
    }

    public void Supprimer(string codeInscription)
    {
        foreach (Etudiant adhérent in Adhérents)
        {
            if (adhérent.CodeInscription == codeInscription)
            {
                Adhérents.Remove(adhérent);
                return;
            }
        }

        throw new Exception("L'étudiant à supprimer n'est pas adhérent !!!");
    }

    public void Supprimer(Etudiant adhérent)
    {
        Supprimer(adhérent.CodeInscription);
    }

    public void Enregistrer()
    {
        string chemin = "Atelier de lecture.bin";
        FileStream fileStream;
        BinaryFormatter binaryFormatter = new BinaryFormatter();

        fileStream = new FileStream(chemin, FileMode.OpenOrCreate, FileAccess.Write);

        binaryFormatter.Serialize(fileStream, Adhérents);

        fileStream.Close();
    }

```

Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 6 8

```

public void Charger()
{
    string chemin = "Atelier de lecture.bin";
    FileStream fileStream;
    BinaryFormatter binaryFormatter = new BinaryFormatter();

    if (File.Exists(chemin))
    {
        try
        {
            fileStream = new FileStream(chemin, FileMode.Open, FileAccess.Read);

            try
            {
                Adhérents =
                    binaryFormatter.Deserialize(fileStream) as List<Etudiant>;
            }
            catch (Exception)
            {
                throw new Exception("Problème de lecture !!!");
            }
            fileStream.Close();
        }
        catch (Exception)
        {
            throw new Exception("Problème d'ouverture !!!");
        }
    }
}

public override string ToString()
{
    string bilan = "Atelier de lecture :\n";

    bilan += string.Format("- Adhérents ({0}) :\n", NombreAdhérents);
    foreach (Etudiant adhérent in Adhérents)
    {
        bilan += string.Format("\t* {0}\n", adhérent);
    }
    return bilan;
}
}

```

3.

```

public partial class Gestion_des_adhérents : Form
{
    Atelier _Atelier = new Atelier();
    public Gestion_des_adhérents()
    {
        InitializeComponent();
    }

    private void Gestion_des_adhérents_Load(object sender, EventArgs e)
    {
        try
        {
            _Atelier.Charger();
            dataGridView_Adhérents.DataSource = _Atelier.Adhérents;
        }
        catch (Exception)
        {
        }
        FormClosing += new FormClosingEventHandler(Gestion_des_Adhérents_FormClosing);
    }
}

```

Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 7 8

```

void Gestion_des_Adhérents_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult résultat =
        MessageBox.Show("Voulez-vous vraiment fermer l'application ?",
                        "Attention",
                        MessageBoxButtons.YesNo,
                        MessageBoxIcon.Question);
    if (résultat == DialogResult.Yes)
        _Atelier.Enregistrer();
    else
        e.Cancel = true;
}

private void button_Supprimer_Click(object sender, EventArgs e)
{
    string codeInscription = textBox_CodeInscription.Text;

    try
    {
        _Atelier.Supprimer(codeInscription);

        dataGridView_Adhérents.DataSource = null;
        dataGridView_Adhérents.DataSource = _Atelier.Adhérents;
    }
    catch (Exception exception)
    {
        MessageBox.Show(exception.Message,
                        "Attention",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
    }
}

private void button_Ajouter_Click(object sender, EventArgs e)
{
    Etudiant etudiant = new Etudiant(textBox_CodeInscription.Text,
                                      textBox_NomComplet.Text);

    if (_Atelier.Ajouter(etudiant))
    {
        dataGridView_Adhérents.DataSource = null;
        dataGridView_Adhérents.DataSource = _Atelier.Adhérents;
    }
    else
        MessageBox.Show("Cet adhérent existe déjà !!!",
                        "Attention",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}

```

Session	Examen de :	Filière	Epreuve de	Variante	Page
Juin 2019	Passage	TDI	Synthèse	V2	Page 8 8