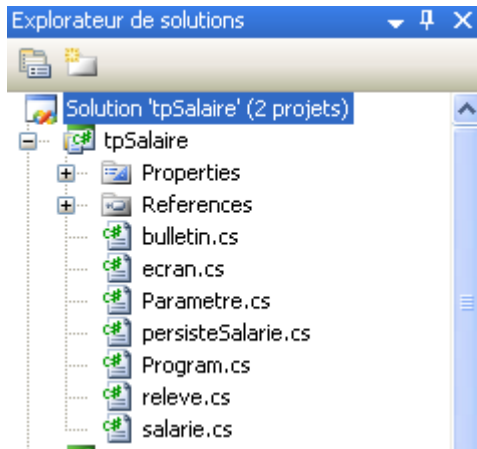


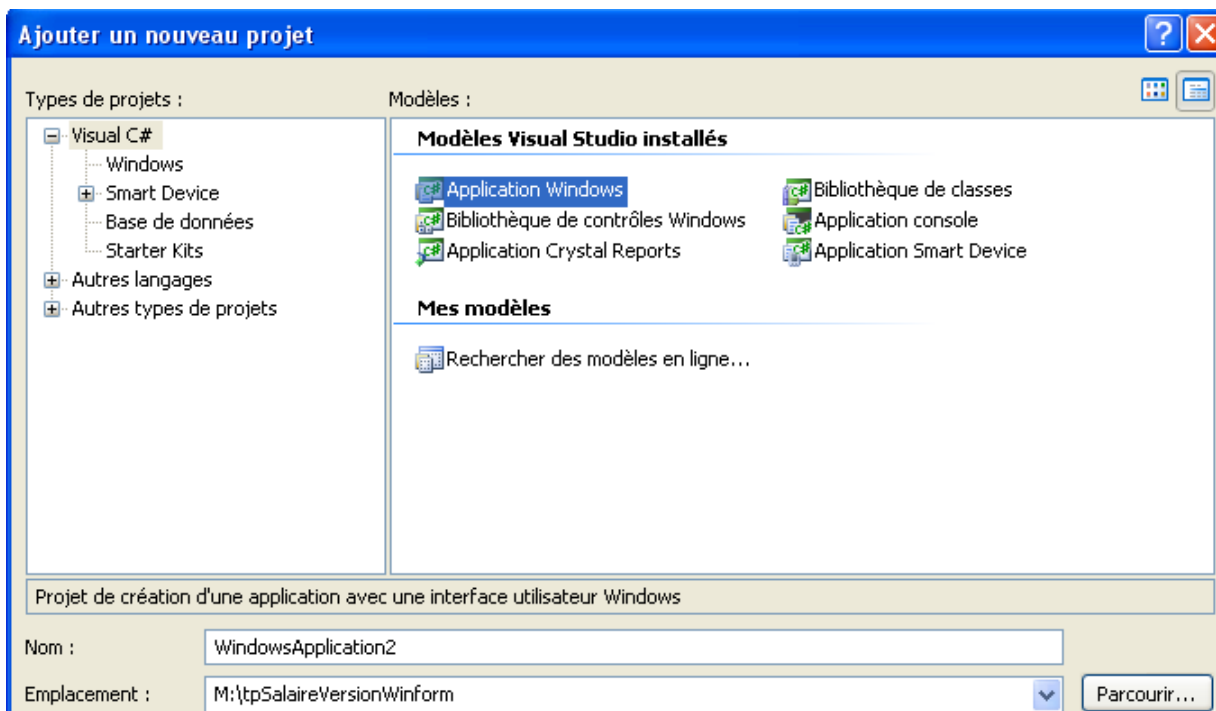
TP Salarie version WinForm

1) Ajout d'un projet WinForm

- Ouvrir le projet *TPSalaire*
- A partir de l'explorateur de solutions, sélectionner la solution :

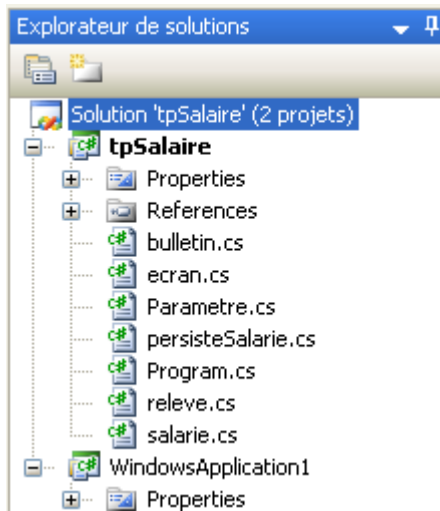


- Click gauche/ajouter/nouveau projet
- Indiquer qu'il s'agit d'un projet Winform (application Windows) :

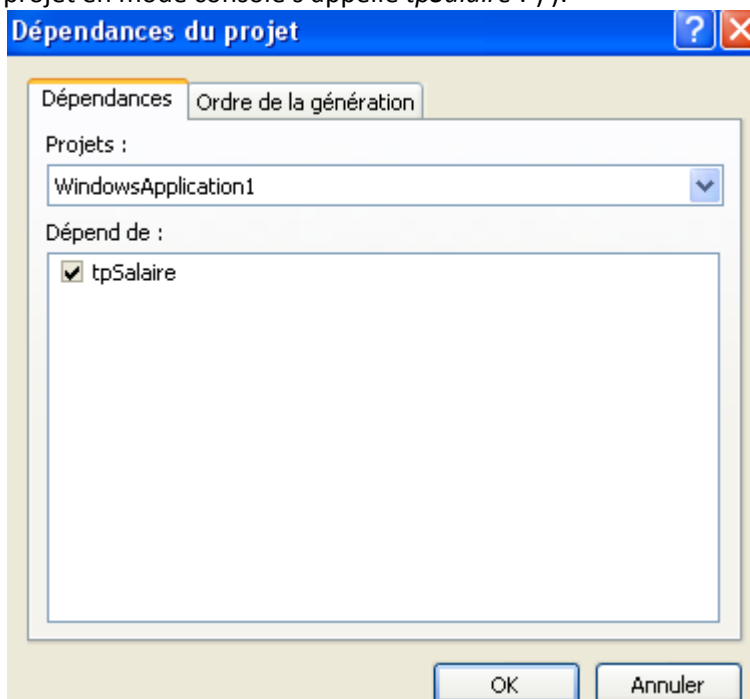


- Indiquer le répertoire du projet (dans le répertoire de votre « solution »)
- Valider
- Votre explorateur de solution vous indique la présence de ce nouveau projet dans la solution :

TP Salarie version WinForm



- Sélectionner ce nouveau projet dans l'explorateur de solution
- Faire un click gauche sur ce projet (WindowsApplication1)
- Sélectionner « définir comme projet de démarrage »
- Click gauche sur ce projet/dépendance du projet, sélectionner le tp salaire (si bien sûr votre projet en mode console s'appelle *tpSalaire* :-)

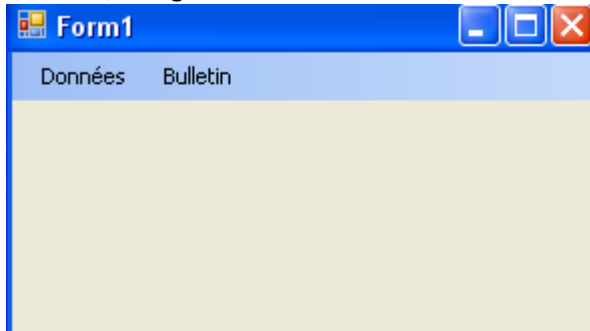


On y est presque, il faudra dans chaque prochain formulaire ajouter : `using tpSalaire` (ceci permet d'accéder directement aux classe sans les préfixer par *tpSalaire*).

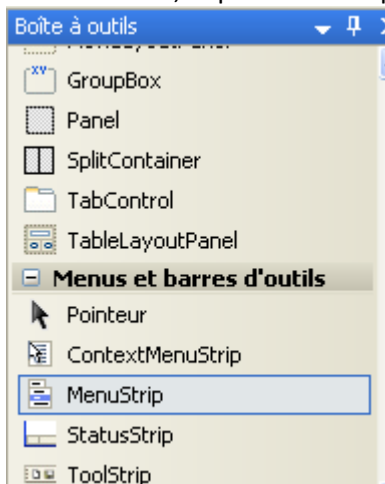
TP Salarie version WinForm

2) Chargement des données

Les données du salariés seront chargées à partir du fichier sérialisé et en activant le menu Données/charger



Pour faire cela, déposer un composant MenuStrip dans le formulaire :



Ajouter les options, *Données/charger* et *Bulletin* dans votre composant de type StripMenu
Lancer l'application et vérifier que ces menus et sous-menus apparaissent.

En cliquant sur le menu Données/Charger une procédure s'ouvre : c'est l'endroit où il faut écrire le code qui sera appelé lorsque l'utilisateur cliquera sur ce sous-menu.

```
private void chargerToolStripMenuItem1_Click(object sender, EventArgs e)
{
}
}
```

Le code à écrire permet de récupérer le salarié à partir du fichier sérialisé :

```
this.leSalarie = PersisteSalarie.charge("salarie.sr");
```

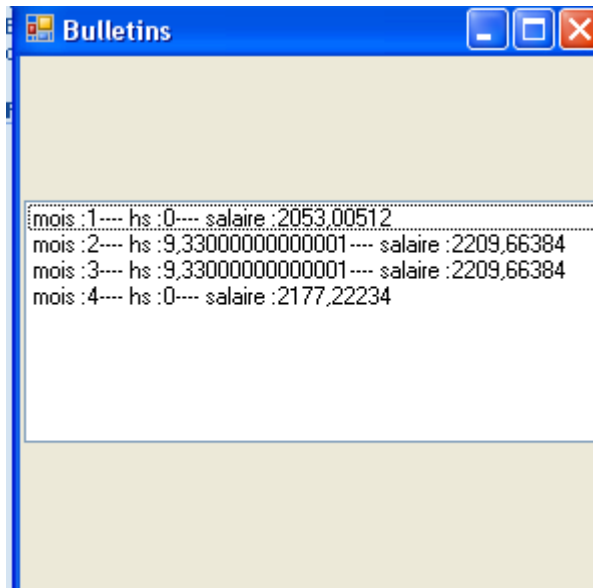
Attention, le fichier sérialisé doit être déplacé dans le répertoire *bin* du projet WinForm

La classe doit donc contenir un attribut privé, trouver lequel ?

3) Premier cas d'utilisation

Il s'agit simplement de visualiser tous les bulletins du salarié :

TP Salarie version WinForm



Ceci se fait dans une ListBox

Pour celà :

- Ajouter un nouveau un nouveau formulaire : click gauche sur le projet/ajouter/formulaire Windows
- Déposer une listBox, nommez-la *lstBulletins*
- Ajouter dans le menu *Bulletins* un sous-menu *voir*
- Ce formulaire doit s'ouvrir lorsque l'on clique sur ce sous-menu *voir* : pour cela, double-cliquez sur ce sous-menu : le gestionnaire de l'événement click apparaît :

```
private void voiToolStripMenuItem_Click(object sender, EventArgs e)
{
}
```

Il faut écrire ici le code qui va permettre d'ouvrir le formulaire *FrmBulletins* **et fournir les informations à ce formulaire pour qu'il puisse afficher les bulletins**, c'est à dire l'objet *leSalarie* déjà construit.

Pour cela, on va utiliser un second constructeur de la classe *FrmBulletins* :

```
private void voiToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmBulletins f = new FrmBulletins(this.leSalarie); // appel de
//ce second constructeur
    f.Show(); // affiche le formulaire créé
}
```

Pour que cet appel soit possible, il faut ajouter un attribut privé *monSalarie* (par exemple) de type *Salarie* dans la classe *FrmBulletins* et ajouter un second constructeur qui valorise ce champ.

```
public FrmBulletins() // constructeur déjà présent
{
    InitializeComponent();
}
```

TP Salarie version WinForm

```
public FrmBulletins(Salarie s) // constructeur à ajouter
{
    InitializeComponent();
    this.monSalarie = s;
}
```

Il ne reste plus qu'à charger la listBox avec les informations du salarié ; ceci doit se faire dès l'ouverture du formulaire FrmBulletins. On peut utiliser le constructeur pour cela :

```
public FrmBulletins(Salarie s)
{
    InitializeComponent();
    this.leSalarie = s;
    this.chargeListe(); // ajout de cette instruction
}
```

Bien sûr, il faut aussi écrire le code de cette méthode :

```
private void chargeListe()
{
    for(int i = 0; i < this.monSalarie.nbBulletins(); i++)
    {
        Bulletin b = this.monSalarie.getBulletin(i);
        this.lstBulletins.Items.Add("mois : " + b.getNumMois().ToString() + "--
-- hs : "+b.nbHS().ToString() + "---- salaire : " +
b.salaireNet().ToString());
    }
}
```

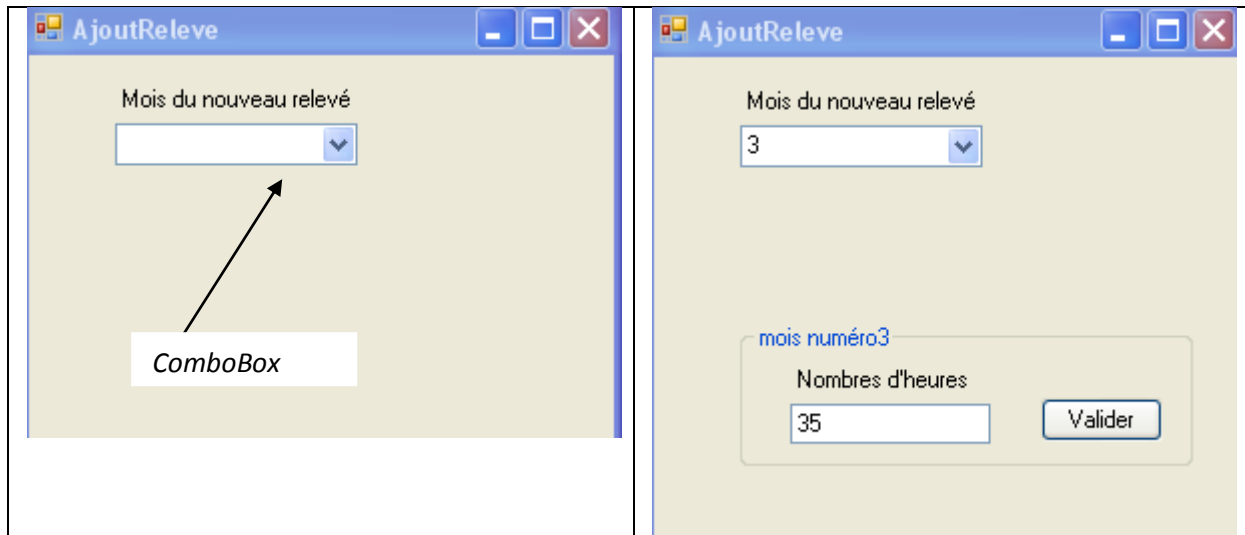
4) Deuxième cas d'utilisation

Cas : ajout d'un relevé

- 1- L'utilisateur demande l'ajout d'un relevé
- 2- Le système retourne le formulaire correspondant
- 3- L'utilisateur sélectionne un mois parmi les mois où figurent des relevés
- 4- Le système retourne un champ de saisie du nombre d'heures pour ce nouveau relevé pour ce mois
- 5- L'utilisateur saisit le nombre d'heures
- 6- Le système enregistre ce nouveau relevé

L'interface peut ressembler à ceci :

TP Salarie version WinForm



Ce formulaire s'ouvrira à partir du premier formulaire dans un sous-menu *Bulletins/ajouter un relevé*

Il faudra procéder de la même manière que pour le premier cas d'utilisation, concernant la construction du nouveau formulaire et l'ajout d'un second constructeur.

```
private void ajouterUnRelevéToolStripMenuItem_Click(object sender,
EventArgs e)
{
    FrmAjoutReleve f = new FrmAjoutReleve(this.leSalarie);
    f.Show();
}
```

Le formulaire doit s'ouvrir avec le *comboBox* chargé avec les numéros de mois des seuls bulletins existants ; ceci se fait dans une méthode *charge Combo* appelée par le constructeur :

```
private void chargerCombo()
{
    for (int i = 0; i < this.monSalarie.nbBulletins(); i++)
    {
        Bulletin b = this.monSalarie.getBulletin(i);
        this.cmbRelevés.Items.Add(b.getNumMois());
    }
}

public FrmAjoutReleve(Salarie s)
{
    InitializeComponent();
    this.monSalarie = s;
    this.chargerCombo();
}
```

Remarque : la classe *Salarie* possède une méthode *nbBulletins()* (à ajouter si nécessaire).

Pour afficher la boîte de saisie pour le mois sélectionné, il faut écrire le code sur l'événement *SelectedIndexChanged* :

TP Salarie version WinForm

```
private void cmbRelevés_SelectedIndexChanged(object sender, EventArgs e)
{
    string mois = this.cmbRelevés.SelectedItem.ToString();
    this.indiceSelectionné = this.cmbRelevés.SelectedIndex;
    this.groupBox1.Visible = true;
    this.groupBox1.Text = "mois numéro" + mois;
}
```

Remarques :

- Il faut déclarer un attribut privé *indiceSelectionné*, car on en aura besoin dans une autre méthode du formulaire. Cet index (commençant à 0) correspond au rang du mois dans l'ArrayList des bulletins du salarié.
- Les composants (textBox, Label et bouton) sont dans un conteneur de type GroupBox qui est mis *visible* à faux à l'ouverture du formulaire.

Pour enregistrer ce nouveau relevé, on écrit le code dans l'événement click du bouton :

```
private void btnValider_Click(object sender, EventArgs e)
{
    int nbHeures = Convert.ToInt32(this.txtNbHeures.Text);
    Bulletin b =
    (Bulletin) this.monSalarie.getBulletin(this.indiceSelectionné);
    b.ajouterReleve(nbHeures);
}
```

Remarque :

On a surchargé la méthode *ajouterReleve* de la classe *Bulletin* :

```
public void ajouterReleve( Double nbHeures)
{
    int numero = this.mesRelevés.Count + 1;
    this.ajouterReleve(numero, nbHeures);
}
```

Ainsi les numéros de bulletins s'incrémentent tout seuls.