

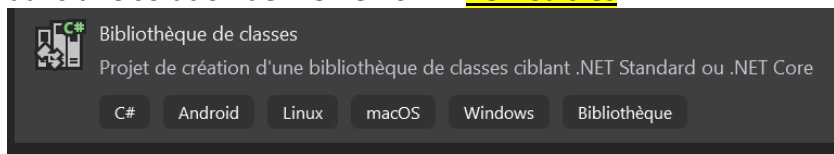
SEQUENCE 4 – STRUCTURES REPETITIVES
SEANCE 3 - VERIFIER DES SAISIES A L'AIDE DES BOUCLES
FAIRE UNE LIBRAIRIE

OBJECTIFS

- Construire sa propre librairie utilitaire pour vérifier les saisies
- Maîtriser les structures : for, while, do...while

EXO 1 : CONSTRUIRE UNE LIBRAIRIE

1. Créez un répertoire « **MesLibrairies** » dans « **R1_01_InitiationDev_C#** » puis au sein de ce répertoire, créez un projet « **VerifSaisies** » de type « **bibliothèque de classes** » dans une solution de même nom « **VerifSaisies** ».



Il n'y aura alors pas de fichier « Program.cs » car il n'y aura pas de main. **Le but est uniquement de définir des méthodes qui seront utiles à tous vos futurs projets.**

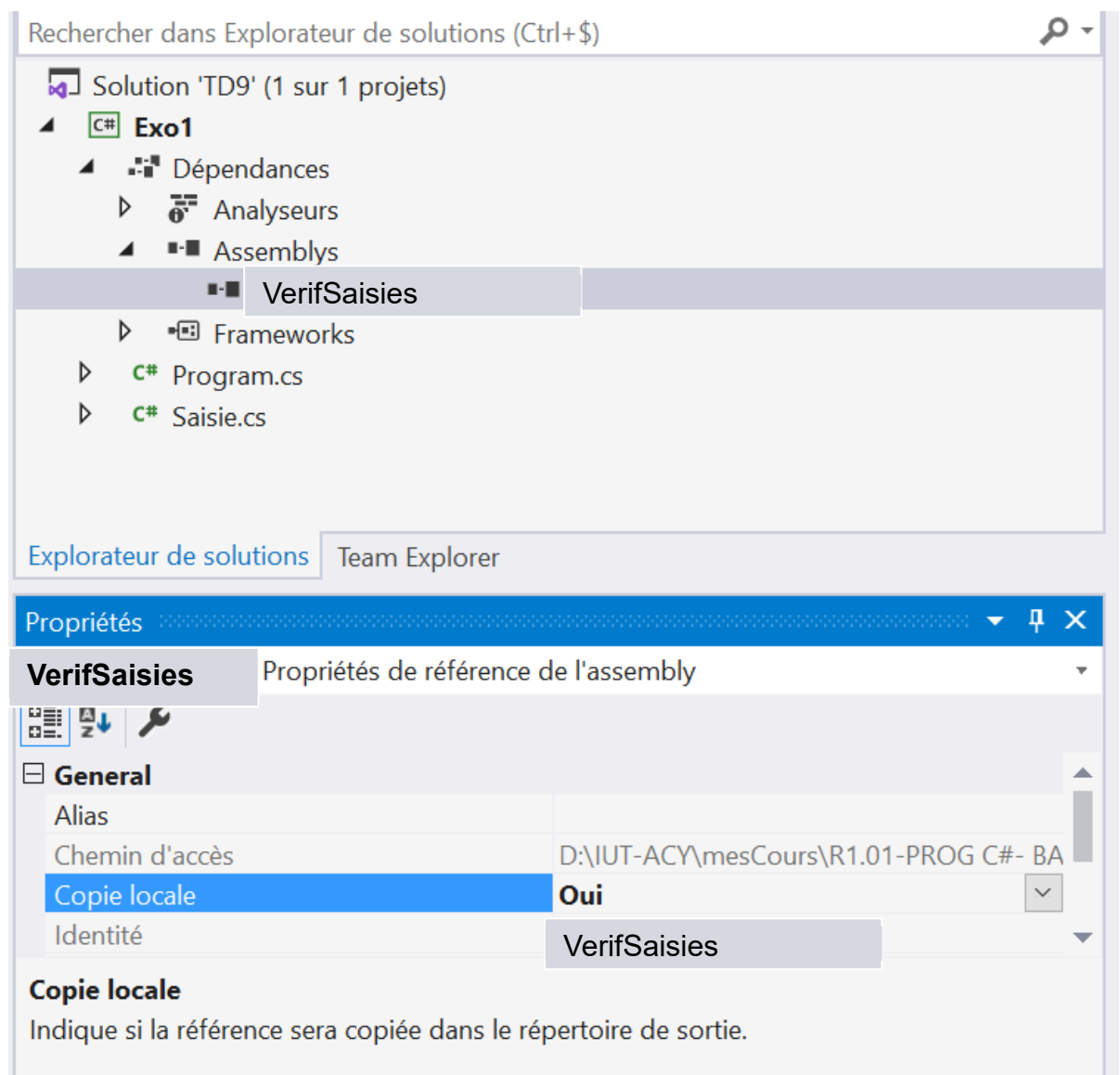
Nommez votre classe « Saisie.cs ».

2. Dans la classe « **Saisie.cs** » placez ce code :

Il a pour but de vérifier que l'utilisateur fait bien une saisie d'un entier. Il évitera ainsi tout problème de conversion et de plantages. Compilez, cela ne lance rien mais **génère une dll : librairie de méthodes compilées prêtes à l'emploi**

```
public static int SaisieInt()
{
    int res;
    bool reussi = int.TryParse(Console.ReadLine(), out res);
    while ( ! reussi )
    {
        Console.WriteLine("Erreur de saisie. Il faut un entier : ");
        reussi = int.TryParse(Console.ReadLine(), out res);
    }
    return res;
}
```

3. Créez un projet « **Exo1_Tests_VerifSaisies_dll** » dans une solution nommée « **Seance3_VerificationsSaisies** » dans votre répertoire « **Se-quence_4_StructuresRepetitives** ». Il va s'appuyer sur votre dll. Au sein de votre projet :
 - Faites une référence à la librairie précédemment créée : clic droit sur les dépendances dans l'explorateur de solution / Ajouter une référence de projet/ puis parcourir afin d'atteindre la **dll** « **VerifSaisies** .dll » générée dans le répertoire **bin/debug** du projet « **VerifSaisies** »
 - Activez l'option Copie Locale
 - N'oubliez pas le « using **VerifSaisies** ; » au sein du fichier « Program.cs »



4. Dans votre « program.cs », utilisez cette méthode pour demander l'âge de l'utilisateur puis affichez un message récapitulatif. Ex : « Vous avez 10 ans ». Essayez de faire des erreurs de saisie.

5. Enrichissez la classe « Saisie », définissez les méthodes et testez les au fur et à mesure dans votre program.cs. Pensez à recompiler la dll.
- **SaisieUInt()** : fait la saisie d'un entier non signé (≥ 0) puis le retourne. Utilisez-la pour la saisie de l'âge.
 - **SaisieDouble()** : fait la saisie d'un nombre réel puis le retourne. Utilisez-la pour demander le poids de l'utilisateur. Testez avec des valeurs comme 5.5 car pour rappel le . faisait planter.
 - **SaisieDoublePositif()** : fait la saisie d'un nombre réel **positif** puis le retourne. Utilisez-la pour remplacer l'appel précédent du le poids de l'utilisateur.
 - une surcharge de la méthode **SaisieDoublePositif** avec 2 paramètres paramètre : double min, double max
 - **SaisieDateTime()** : fait la saisie d'une date au format JJ/MM/AAAA et renvoie un objet DateTime. Utilisez-la pour demander la date de naissance de l'utilisateur.
 - **SaisiePlaquelImmatriculation()** : format attendu : AA-000-AA, composé de sept caractères alphanumériques : deux lettres, trois chiffres et deux lettres, ces trois groupes étant séparés par des tirets
 - **SaisieUnCaracParmiDeuxChoix** (String message, String choix1, String choix2)

EXO 2 : REUTILISER UNE LIBRAIRIE

1. Ajoutez un projet « **Exo2_Utilisation_Verif_Saisies** » à votre solution. Votre programme doit faire la saisie :

- Du statut de l'employé : C pour cadre, O pour ouvrier
- Du nombre d'enfant(s) de l'employé : chiffre compris entre 0 et 20.

Rem : Vous **devez** réutiliser les méthodes précédemment définies en **ajoutant la librairie à votre projet.**

```
-----
Entrez categorie(O : ouvrier - C : cadre):
r
Erreur de saisie
-----
Entrez categorie(O : ouvrier - C : cadre):
e
Erreur de saisie
-----
Entrez categorie(O : ouvrier - C : cadre):
0
Entrez le nombre d'enfant compris entre 0 et 20:
23
Erreur de saisie
Entrez le nombre d'enfant compris entre 0 et 20:
3
```

2. Votre programme doit ensuite afficher un récapitulatif tel que :

- s'il a des enfants : *L'employé Dupont est un _____ * ayant _____ enfant(s)*
 - s'il n'a pas d'enfant : *L'employé Rabah est un _____ * n'ayant pas d'enfant*
- *. doit mentionner ouvrier ou bien cadre

REM : Attention, vous devez optimiser ! Vous pouvez utiliser Console.Write qui évite de sauter une ligne et permet d'écrire le résumé en 2 temps. Vous devez exécuter 4 jeux de tests pour valider correctement ce programme :

Données en entrée			Données en sortie
Categorie	Nb enfants	Nom	Message
O	0	Delate	<i>L'employé Delate est un ouvrier n'ayant pas d'enfant.</i>
O	5	Fabre	<i>L'employé Fabre est un ouvrier ayant 5 enfant(s)</i>
C	0	Trau	<i>L'employé Trau est un cadre n'ayant pas d'enfant.</i>
C	3	Badun	<i>L'employé Badun est un cadre ayant 3 enfant(s)</i>