

Classe Extension

Les classes d'extension permettent d'ajouter de nouvelles méthodes à des classes ou interfaces existantes, sans avoir besoin de les modifier ou de les hériter. Cela est particulièrement utile dans les cas suivants :

1. Étendre des classes ou des interfaces fermées :

- Si vous ne pouvez pas modifier le code source d'une classe (ex. : classes intégrées comme `String`, `List<T>`), les classes d'extension permettent d'ajouter des fonctionnalités.

2. Centraliser la logique commune :

- Réutilisez des fonctions dans plusieurs parties du projet sans dupliquer le code.

3. Améliorer la lisibilité et la fluidité :

- Les méthodes d'extension s'utilisent comme si elles faisaient partie de la classe d'origine, ce qui rend le code plus naturel et fluide.

Classes partielles : Utilisées pour organiser le code d'une classe en plusieurs fichiers (base, calculs, affichage).

Classes d'extension : Ajoutent des méthodes sans modifier le code source (validation et calcul du bonus).

Une classe d'extension doit être :

- Une classe statique.
- Contenir des méthodes statiques.
- La première paramètre de chaque méthode doit être `this` suivi du type à étendre.

```
public static class EmployeExtensions
{
    // Vérifie si les propriétés essentielles de l'employé sont valides
    // GestionEmployes nom de votre namespace

    public static bool EstValide(this GestionEmployes.Employe employe)
    {
```

```

        return !string.IsNullOrEmpty(employee.Nom) && employee.SalaireMensuel > 0;
    }

    // Ajoute un bonus en fonction du poste
    ///
    public static double CalculerBonus(this GestionEmployes.Employe employe)
    {
        if (employee.Poste.ToLower() == "manager")
            return employee.SalaireMensuel * 0.2; // 20% pour les managers
        if (employee.Poste.ToLower() == "développeur")
            return employee.SalaireMensuel * 0.1; // 10% pour les développeurs
        return 0; // Pas de bonus pour les autres postes
    }
}

```

Mise à jour du programme principal

Fichier Program.cs

Ajoutez du code pour tester toutes les nouvelles fonctionnalités.

```

static void Main(string[] args) {
    Employe employe =
        new Employe { Id = 1, Nom = "Ahmed B.", Poste = "Développeur", SalaireMensuel = 2500,
            Prime = 3000 };
    if (employe.EstValide()) {
        // Afficher les détails complets employe.AfficherDetailsComplets();
        // Calculer et afficher le bonus
        double bonus = employe.CalculerBonus();
        Console.WriteLine($"Bonus pour le poste '{employee.Poste}': {bonus} €");
    } else { Console.WriteLine("Les informations de l'employé sont invalides."); }
}
}

```