

## Module 2

### La Classe et l'Objet

---

Objectif: passer d'un code séquentiel à des objets qui regroupent état et comportements.

## Changelog – V0.0.2

---

- Deck séparé sur classe, objet, constructeur et visibilité.

## Pourquoi une classe ?

---

- Une classe modèle un concept du domaine.
- Un objet est une instance concrete.
- `new` construit cet objet en mémoire.

```
Produit clavier = new Produit("Clavier", 49.99);
```

## Anatomie d'une classe

---

```
public class Produit
{
    private string _nom;

    public string Nom
    {
        get => _nom;
        set => _nom = value;
    }

    public double Prix { get; set; }

    public Produit(string nom, double prix)
    {
        _nom = nom;
        Prix = prix;
    }
}
```

## Champs, propriétés, constructeur

---

- Le champ stocke une valeur interne.
- La propriété contrôle l'accès.
- Le constructeur impose un état initial valide.

## Méthodes et mot-clé `this`

---

```
public void AppliquerRemise(double pourcentage)
{
    this.Prix = this.Prix - (this.Prix * pourcentage / 100);
}
```

- Une méthode a un nom, des paramètres et un type de retour.
- `this` désigne l'objet courant.

## public vs private

---

- `public` expose l'API d'une classe.
- `private` protège les détails internes.
- Une bonne visibilité limite les usages incohérents.

## Checklist de sortie

---

- Savoir lire et écrire une classe simple.
- Savoir instancier un objet avec `new`.
- Différencier champ, propriété, constructeur et méthode.