Programmation, bases de données et serveurs-AEC(LEA.D4)
Technique de l'informatique DEC Intensif(420.B0)

Introduction à la programmation Orientée Objet
420-W20-SF
Pondération: 3-3-3
Hiver 2020

Professeur :
André Boumso

Cégep de Sainte-Foy
Direction de la formation continue

Objectifs

- À la fin de cette leçon, l’étudiant sera en mesure de dire:
  - ce qu'est la programmation orientée objet
  - Ce qu’est une classe, un objet
  - Ce qu’est une méthode et comment l’implémenter,
  - Ce qu’est un diagramme de classe et le concevoir,

Leçon 3: La programmation Orientée Objet

Présentation de la POO

- La programmation orientée objet est une approche du développement logiciel dans laquelle la structure du logiciel est basée sur des objets interagissant les uns avec les autres pour accomplir une tâche.
- Un objet représente, de façon **abstraite,** n'importe qu'elle entité de notre environnement.
- Un objet est une structure pour incorporer des données et les opérations de traitements de ces données.

Rappels

- Qu'est qu'un type de donnée Structure?
- Qu'est qu'une fonction?

CLASSES

- Une classe est une collection d'un nombre fixe de composants. Les composants d'une classe sont appelés les membres de la classe.
- Les membres d'une classe sont:
  - Les données (attributs)
  - Les opérations (fonctions)
- Une classe ressemble à une structure mais avec les données membres et les fonctions définies dans la classe.

LES CLASSES

- La classe représente un modèle, qui décrit les différents états et comportements de certains objets.
- Un diagramme de classes peut vous aider à visualiser les attributs et les opérations d'une classe.

ÉLÉMENTS D'UNE CLASSE

- Une classe comprend les éléments suisvants:
  - Une déclaration: `public class NomDeLaClasse`
  - Un corps compris entre deux accolades:

```
public class NomDeLaClasse
{
 • // … Le corps de classe va ici…
}
```

  - Un constructeur: on verra plus tard
  - Des champs/attributs: ce sont les données (variables) définies dans la classe.

ÉLÉMENTS D'UNE CLASSE

- Les champs:
  - `private string nom;`
- Les propriétés:
  - `private string Nom { get; set; }`
- Les méthodes: ce sont des fonctions définies à l'intérieur de la classe.

```
public class Personne
{
      // Déclaration de champ/attribut d'une donnée membre
      private string m_nom;
      private string m_prenom;
      private int m_age;


      // Constructeur par défaut
      public Personne()
      {
      }
      // Autre déclaration de constructeur
      public Personne(string p_nom)
      {
            this.m_nom = p_nom;
      }


      // Déclaration d'une propriété
      public string Nom
      {
            get { return m_nom; }
            set { m_nom = value; }
      }
      // Déclaration de Méthode
      public static void Ecrire()
      {
      Console.WriteLine("{0} a dit: J'aime la prog!",
      m_nom);
      }
}
```

Objets

- Un objet est une variable de type Class.
  - `Personne personne;`
- La création d'un objet s'appelle instanciation de l'objet.
  - `Personne personne = New Personne();`
  - `Personne personne = New Personne(params);`

Le mot réservé "this"

- Le mot réservé this en C # est utilisé pour référencer l'objet courant, quand il est utilisé à partir d'une méthode de la même classe.
  - `this.unCham; //Accès à un champs de la classe`
  - `this.uneMethode; //Accès à une méthode`
  - `this(1,2);  //Accès au constructeur`

```
public personne CreerPersonne(string p_nom,string p_prenom, int p_age)
{

     this.m_nom = p_nom;
     this.m_prenom = p_prenom;
     this.m_age = p_age;

     return this;
}
```

Diagramme de classes

- Le Langage Unifié de Modélisation (UML) permet de représenter les classes sous forme de diagrammes.
- Dans la notation UML, une classe est représentée comme un rectangle ayant:
  - Un nom
  - Des attributs (membres de la classe)
  - Des opérations (méthodes)

Diagramme de classes

Personne
- m_nom: string
- m_prenon: string
- m_age: int
+ ECrire(): void

Modélisation Orientée Objet

- Elle permet de déterminer l'ensemble de tous les objets impliqués dans la résolution d'un problème.
- Elle consiste:
  - L'identification des classes,
  - L'identification des attributs de classes,
  - Celle des opérations sur les classes,
  - Et les relations entre les classes.

Le Principe SRP

- C'est le premier principe d'un de principes dits SOLID, qui régissent un ensemble de bonnes pratiques de développement en POO.
- SRP: Single Responsability Principle
  - Principe de la Responsabilité Unique
    - Une classe doit avoir une seule responsabilité pour éviter le couplage,
    - Si deux tâches sont effectuées dans classe, la scinder.

Le Principe SRP

```
class Client
{
    void Ajouterbd(Database db)
    {
        try
        {
            db.Ajouter();
        }
        catch (Exception ex)
        {
            File.WriteAllText(@"C:\Error.txt", ex.ToString());
        }
    }
}

class Client
{
    private FileLogger logger = new FileLogger();
    void Ajouter(Database db)
    {
        try
        {
            db.Add();
        }
        catch (Exception ex)
        {
            logger.Handle(ex.ToString());
        }
    }
}
class FileLogger
{
    void Handle(string error)
    {
        File.WriteAllText(@"C:\Error.txt", error);
    }
}
```

Source: https://blog.scottlogic.com/2018/06/26/solid-principles.html