OFPPT
La voie de l'avenir

DÉVELOPPEMENT DIGITAL

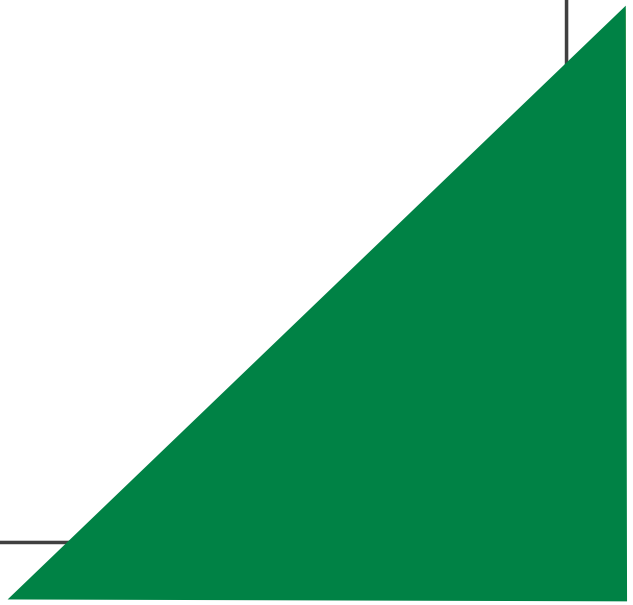
PROGRAMMER EN ORIENTÉ OBJET

CHAPITRE 4

Principe de l'abstraction Et les interfaces

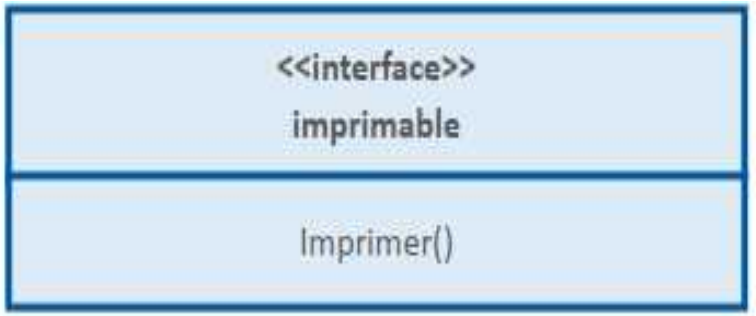
1- l'abstraction

2- les interfaces



Manipuler les interfaces : Définition et Utilité

- Comme une classe et une classe abstraite, une interface permet de définir un nouveau type (référence).
- Une interface est une forme particulière de classe où **toutes les méthodes sont abstraites**.



Utilité des interfaces

Les interfaces permettent de :

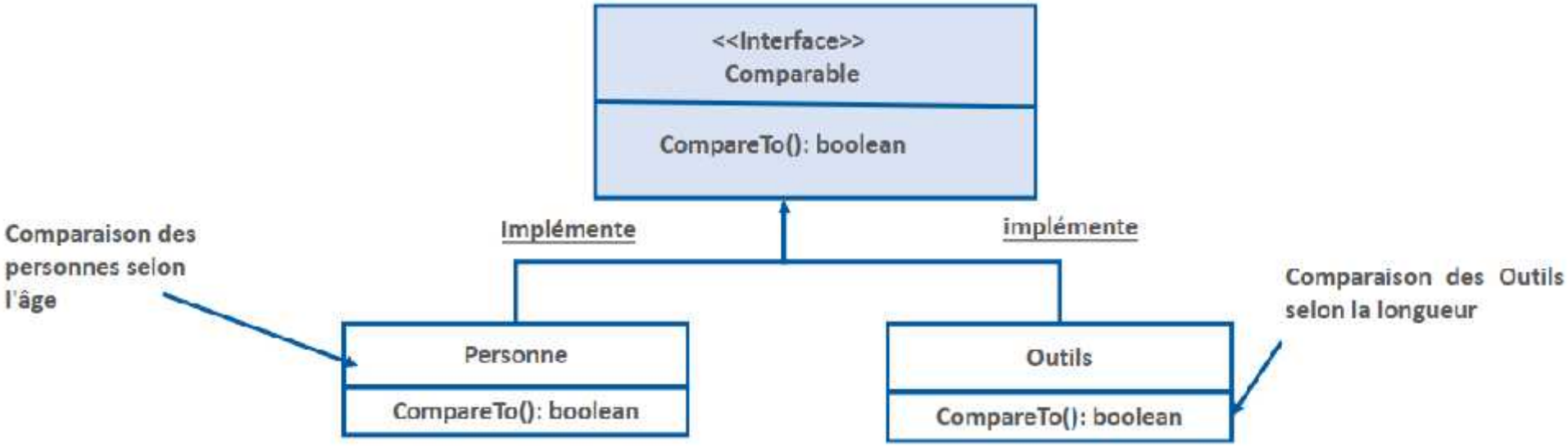
- Spécifier des propriétés qui peuvent être utilisées par les classes qui implémentent ces interfaces.
- Obliger les classes qui les implémentent de définir les méthodes abstraites déclarées dans les interfaces.
- Tirer profit du polymorphisme avec des instances dont les classes ne font pas partie de la même hiérarchie d'héritage.

Manipuler les interfaces : Implémentation

- On dit qu'une **classe implémente une interface**, si elle fournit une implémentation (c'est-à-dire un corps) pour chacune des méthodes abstraites de cette interface.
- Si une classe implémente plus d'une interface, elle doit implémenter toutes les méthodes abstraites de chacune des interfaces.

Exemple1 :

- Si l'on souhaite caractériser la fonctionnalité de comparaison qui est commune à tous les objets qui ont une relation d'ordre (plus petit, égal, plus grand), on peut définir l'interface Comparable.
- Les classes Personne et Outils qui implémentent l'interface Comparable **doivent présenter une implémentation de la méthode CompareTo() sinon elles seront abstraites.**



Manipuler les interfaces : Implémentation

Exemple 2 :

- Supposons que nous voulions que les classes dérivées de la classe Forme disposent toutes d'une méthode imprimer() permettant d'imprimer les formes géométriques.

Solution 1:

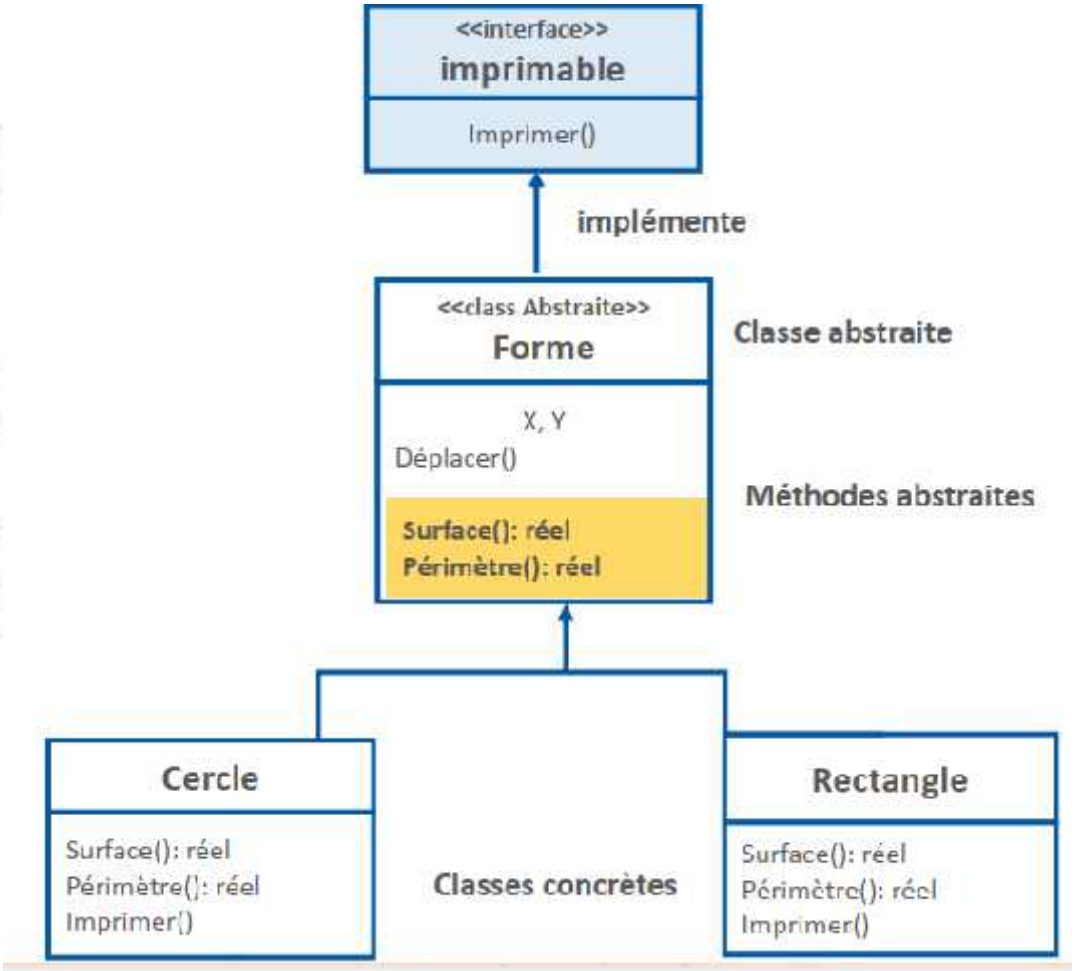
- Ajouter une méthode abstraite Imprimer()à la classe Forme et ainsi chacune des sous-classes concrètes devrait implémenter cette méthode.

→ Si d'autres classes (qui n'héritent pas de Forme souhaitaient également disposer des fonctions d'impression, elles devraient à nouveau déclarer des méthodes abstraites d'impression dans leur arborescence.

Solution 2 :

La classe forme implémente l'interface imprimable

- ayant la méthode Imprimer()



Manipuler les interfaces : Implémentation

```
from abc import ABC , abstractmethod
```

```
class Imprimable(ABC):
    @abstractmethod
    def imprimer(self):
        pass
class Forme(Imprimable):
    def __init__(self, n = "nom"):
        self._Nom= n
    @ property
    def Nom(self):
        return self._Nom
    @Nom.setter
    def Nom(self, n):
        self._Nom = n
    def imprimer(self):
        pass
    def __str__(self):
        return f"Nom : {self.Nom}"
    @abstractmethod
    def Surface(self):
        pass
```

```
class Rectangle(Forme):
    def __init__(self, n="nom", l=0 , L=0):
        super().__init__(n)
        self.__L = L
        self.__l=l
    @property
    def L(self):
        return self.__L
    @L.setter
    def L(self , l):
        self.__L =l
    @property
    def l(self):
        return self.__l
    @l.setter
    def l(self,l):
        self.__l=l
    def Surface(self):
        return self.L*self.l
    def imprimer(self):
        print("Impression Rectangle")
    def __str__(self):
        return f"{super().__str__()} Longueur : {self.l} Largeur : {self.L}"
```

Manipuler les interfaces : Implémentation

```
class Cercle(Forme):
    def __init__(self, n="nom",r=0):
        super().__init__(n)
        self.__R= r
    @property
    def R(self):
        return self.__R
    @R.setter
    def R(self , r):
        self.__R =r
    def imprimer(self):
        print("Impression cercle")
    def __str__(self):
        return f"{super().__str__()} Rayon : {self.R}"
    def Surface(self):
        return math.pi*math.pow(self.R,2)
```

Programme Principale :

```
R = Rectangle("rectangle", 5,9)
R.imprimer()
print(R.Surface())
C= Cercle("cercle",8)
C.imprimer()
print(C.Surface())
```

CHAPITRE 5

Les interfaces graphiques

1- Installation des bibliothèques externes

2- la bibliothèque Tkinter

Installation des bibliothèques externes

Bibliothèque en Python

- Dans Python, une bibliothèque est un ensemble logiciel de modules (classes (types d'objets), fonctions, constantes...) ajoutant des possibilités étendues à Python : calcul numérique, graphisme, programmation internet ou réseau, formatage de texte, génération de documents, etc.
- Il en existe un très grand nombre, et c'est d'ailleurs une des grandes forces de Python.
- La plupart est regroupée dans PyPI (Python Package Index) le dépôt tiers officiel du langage de programmation Python.

Bibliothèque standard

- La distribution standard de Python dispose d'une très riche bibliothèque de modules étendant les capacités du langage dans de nombreux domaines.
- La bibliothèque standard couvre un large éventail de fonctionnalités, notamment :
  - Modules de date et d'heure
  - Modules des interfaces graphiques
  - Modules numériques et mathématiques
  - Modules de système de fichiers
  - Modules de système d'exploitation
  - Modules pour la lecture et l'écriture de formats de données spécifiques tels que HTML, XML et JSON
  - Modules pour l'utilisation de protocoles Internet tels que HTTP, SMTP, FTP, etc.
  - Modules pour l'utilisation de données multimédias telles que les données audio et vidéo

Installation des bibliothèques externes

Pip (Python Installer Package)

- Pip (Python Installer Package) est le manager de package pour Python
- Pip est un moyen d'installer et de gérer des packages et des dépendances supplémentaires qui ne sont pas encore distribués dans le cadre de la version standard du package
- Pip package est intégré dans l'installation du Python depuis les versions 3.4 pour Python3 et les versions 2.7.9 pour Python2, et utilisé dans nombreux projets du Python.
- En exécutant la commande ci-dessus, il est possible de vérifier que pip est disponible ou non

```
pip --version
```

- **Résultat de l'exécution :**

```
pip 21.2.4 from C:\Users\DELL\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-
packages\Python39\site-packages\pip (python 3.9)
```

- La sortie permet d'afficher la version de **pip** dans votre machine, ainsi que l'emplacement de votre version du Python
- Si vous utilisez une ancienne version de Python qui n'inclut pas **pip**, vous devez l'installer

Installation des bibliothèques externes

- La mise à jour des bibliothèques existantes doit être demandée explicitement :

```
py -m pip install --upgrade SomePackage
```

- Pour désinstaller une bibliothèque, il faut utiliser la commande suivante

```
py -m pip uninstall sampleproject
```

Installation des bibliothèques externes

Bibliothèques graphiques

- La bibliothèque **matplotlib** (et sa sous-bibliothèque pyplot) sert essentiellement à afficher des graphismes. Son utilisation ressemble beaucoup à celle de Matlab.
- Pour installer **matplotlib**, il faut taper la commande :

```
py -m pip install matplotlib
```

```
PS C:\Users\DELL> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.4.3-cp39-cp39-win_amd64.whl (7.1 MB)
     |████████████████████████████████| 7.1 MB 2.2 MB/s
Collecting numpy>=1.16
  Using cached numpy-1.21.3-cp39-cp39-win_amd64.whl (14.0 MB)
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.3.2-cp39-cp39-win_amd64.whl (52 kB)
     |████████████████████████████████| 52 kB 84 kB/s
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from
 matplotlib) (2.4.7)
Collecting pillow>=6.2.0
  Downloading Pillow-8.4.0-cp39-cp39-win_amd64.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 1.7 MB/s
Collecting python-dateutil>=2.7
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     |████████████████████████████████| 247 kB 2.2 MB/s
Requirement already satisfied: six in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from cycler>=0.10
->matplotlib) (1.16.0)
```

Installation des bibliothèques externes

Bibliothèques graphiques

- Tkinter est un module intégré à Python pour développer des applications graphiques.
- Ce module se base sur la bibliothèque graphique Tcl/Tk.
- Pour installer Tk, il faut taper la commande :

```
py -m pip install tk
```

```
PS C:\Users\DELL> pip install tk
Collecting tk
  Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\DELL\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\python.e
xe -m pip install --upgrade pip' command.
```

Créer des interfaces graphiques

Tkinter

- Le programme ci-dessous montre le principe de base de tkinter