Programmation Objet
Séance 1

Arnaud Sallaberry

arnaud.sallaberry@univ-montp3.fr

Objectifs

- Se familiariser aux concepts de la programmation par objets
- Présenter par la pratique quelques problématiques du développement des applications de nouvelle génération
- Devenir et demeurer un interlocuteur actif et compétent :
  - auprès des équipes informatiques
  - auprès des prescripteurs

Rappels Python

- Fichier : `monFichier.py`
- En-tête : `#!/usr/bin/env python`
- Import : `from NomFichier import *`
- Variable :

```
p = 3
p = ‘’un petit texte’’
```

- Fonction : `def maFonction(parametre1):`
- Fonction print :

```
print (‘’un petit texte’’)
print (10)
```

Rappels Python

- Liste :

```
l1 = [12, 2, 2, 6, 8],
l2 = [‘’un’’, ‘’petit’’, ‘’texte’’]
```

- For :

```
for i in l1:
   print i
for i in range(len(l1)):
   print l1[i]
```

Programmation Objet

Pourquoi ?

- Faciliter la réutilisation du code
- Encapsulation
- Difficile de prévoir toutes les utilisations d'un programme

Programmation Objet

Définition d'une structure de données (objet) : permet d'avoir des types composites

- Exemple d'un compte en banque

  Titulaire: chaine de caractères

  Solde: nombre

```
Class Compte:
    « definition d'un compte en banque »
    titulaire = ''''
    solde = 0
```

- `Titulaire` et `solde` sont des **champs** de la classe `Compte` (variables « dans la classe »)

Programmation Objet

- Peut être mis dans une variable :

```
c1 = Compte()
c1.titulaire = ''Sallaberry''
c1.solde = 1000
c2 = Compte()
c2.titulaire = ''Richomme''
c2.solde = 2000
```

- c1 et c2 sont des **instances** de la classe Compte

Programmation Objet

Composition d'objets : Objets dont les champs sont des objets

```
p1 = Personne()
p1.nom = ''Sallaberry''
p1.prenom = ''Arnaud''

c1 = Compte()
c1.titulaire = p1
c1.solde = 100000
```

Programmation Objet

- Des objets en argument de fonctions :

```
def deposer(c,s):
    c.solde = c.solde + s
```

- Erreur si objet non conforme, si `c` et `s` n'ont pas le même rôle dans cette fonction

Programmation Objet

- **Méthode** : fonction dans une classe

```
Class Compte:
   « definition d'un compte en banque »
   def deposer(self,s):
      self.solde += s
```

- Pour fixer la structure:
  - Méthode « constructeur »

Programmation Objet

Une méthode particulière : le **constructeur**

- Créer des instances avec un contenu
- Méthode nommée `__init__`
- Paramètre `self`
- Autant d'arguments/paramètres que l'on veut
- Utilisation de valeurs par défaut
- Appelée lors de la création de l'instance

Programmation Objet

Exemple de constructeur de la classe `Personne`

```
class Personne:
    def __init__(self, n, p):
        self.nom = n
        self.prenom = p
```

Pour créer une instance:

```
p = Personne("Shakespeare","William")
```