

../../../../Latex/_latex_assets/logo_iogs.png

Outils Numériques pour l'Ingénieur·e en Physique

2023-2024 / FISA

6N-099-PHY / ONIP-2

Prog. Objet (100%)

Concepts étudiés

[NUM] Classes et objets

Mots clefs

Python; Programmation objet; Classes;
Objets; Méthodes; Attributs

Sessions

- 0 Cours(s) - 1h30
- 0 TD(s) - 1h30
- 4 TD(s) Machine - 2h00
- 0 TP(s) - 4h30

Travail

Par binôme

Institut d'Optique

Graduate School, France

<https://www.institutoptique.fr>

GitHub - Digital Methods

<https://github.com/IOGS-Digital-Methods>

Introduction à la programmation orientée objet

Dans le cadre du module **ONIP-2**, vous serez amenés à découvrir les concepts de la **programmation orientée objet** et à développer un mini-projet selon ces nouvelles règles.

Dans le projet proposé, on cherchera à **calculer la carte d'éclairement** produit par un **ensemble de sources incohérentes**.

Les sources seront modélisées de manière approchée (valable si l'on n'est pas trop près du composant) comme une **source ponctuelle** ayant un diagramme de rayonnement possédant une symétrie de révolution autour d'un axe.

Aucune fonction ne devra être utilisée en dehors d'un objet.

Déroulement du bloc

Ce module se déroule sur **4 séances** :

Séance 1 Découverte de la programmation objet

Séances 2 et 3 Réalisation du mini-projet en binôme

Séance 4 Evaluation du mini-projet en binôme

Livrables attendus

Afin de faciliter la réalisation du mini-projet proposé, nous vous suggérons tout au long du développement de mettre à jour les documents suivants :

1. **Diagramme de classe** et répartition du travail
2. **Classes commentées** (selon la norme PEP 8) pour générer des objets
3. **Graphiques légendés** incluant toutes les données nécessaires à la bonne compréhension des données présentées
4. **Analyse des figures** obtenues

Ressources

Cette séquence est basée sur le langage Python.

Vous pouvez utiliser l'environnement **Spyder 5** inclus dans *Anaconda 3*.

Des tutoriels Python (et sur les bibliothèques classiques : Numpy, Matplotlib ou Scipy) sont disponibles à l'adresse :

<http://lense.institutoptique.fr/python/>.

Séance 1 - Programmation orientée objet

Dans ce module, vous serez amenés à développer une application selon les principes de la programmation orientée objet.

Afin de vous familiariser avec les principes de base, la première séance sera consacrée à l'étude et la mise en œuvre d'exemples de la programmation orientée objet en Python : écriture d'une classe, instanciation d'un objet, interaction entre les objets.

Acquis d'Apprentissage Visés

A travers les exemples proposés, vous serez capables de :

CÔTÉ NUMÉRIQUE

1. **Créer des classes** incluant des méthodes et des attributs
2. **Instancier des objets** et les faire interagir
3. **Définir et documenter les méthodes et attributs** de chaque classe

Exemples à analyser et exercices

L'ensemble des documents du module ONIP-2 se trouve sur le site du LEnsE : <http://lense.institutoptique.fr/ONIP/> . Les exemples pour cette première séance se trouvent dans la rubrique **BLOC 4**.

Exercice 1 - Classe Point

En vous inspirant de la définition et de l'exemple d'utilisation de la classe **Animal** (simple) :

- créez un nouveau fichier .py
- définissez une classe **Point**, permettant de modéliser un point dans un espace en 2 dimensions par ses coordonnées x et y
- instanciez deux objets de type **Point** avec des coordonnées différentes

En vous inspirant de la définition et de l'exemple d'utilisation de la classe **Animal** (redéfinition str) :

- redéfinissez la méthode `__str__` pour qu'elle affiche les coordonnées d'un objet de type **Point**
- vérifiez vos différentes méthodes

Exercice 2 - Classe Rectangle

Dans le fichier précédent et en utilisant la classe **Point** :

- définissez une classe **Rectangle**, permettant de modéliser un rectangle à partir de deux objets de type **Point** (sommets opposés du rectangle)
- définissez des méthodes **perimetre** et **surface** permettant de calculer le périmètre et la surface d'un objet de type **Rectangle**
- testez l'ensemble de vos méthodes sur différents objets de type **Rectangle**

Exercice 3 - Classe Cercle

Même exercice avec un cercle défini par son centre et un point du rayon.

Outils Numériques

Fonctions et bibliothèques conseillées :

- **Numpy** gestion de matrices
- **Matplotlib** affichage de données
- **Scipy** fonctions scientifiques

Fichiers d'exemple

Classe **Animal** (simple) :

onip_b4_a_classe_simple.py

Classe **Animal** (redéfinition str) :

onip_b4_b_classe_simple_redéfinition.py

Classes **Dog** et **Cat** :

onip_b4_c_classe_heritage