



Bachelier en Informatique de Gestion

Programmation Orientée Objet

Enseignement supérieur économique de type court

Code FWB : 7525 21 U32 D3

Code ISFCE : 4IPO3



= BUROTIX ()

Le Professeur, c'est qui ?



Alain Wafflard

isfce.aw@gmail.com



Ingénieur civil

25 ans dans le privé
Processus d'entreprise
Automatisation, informatisation



Enseignant

ISFCE Etterbeek
mais auparavant : IFOSUP Wavre, IFC Ixelles,
IEPSCF Evere, IPAM La Louvière, EPFC
Bruxelles



Activité complémentaire

burotix.be
formation en tableur
conception de base de données
développement de templates



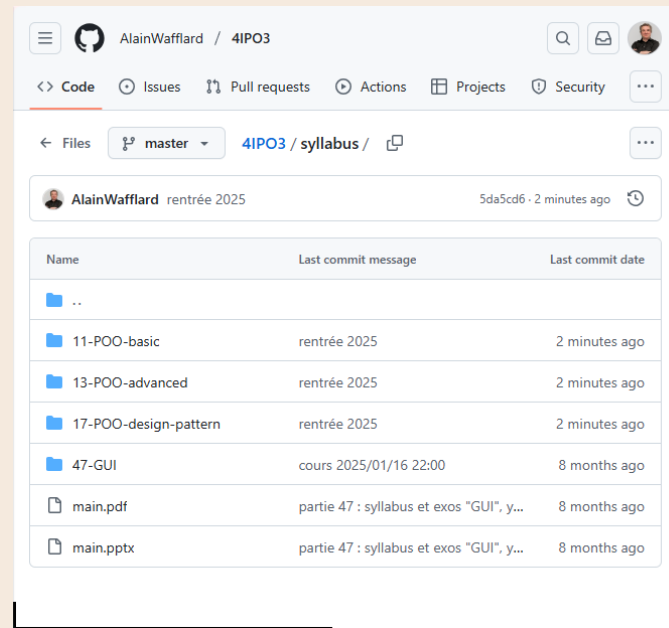
Acquis d'Apprentissage

- Seuil de réussite
 - concevoir, installer et utiliser des objets appropriés à la solution
 - concevoir et mettre en œuvre une procédure de test
 - justifier sa méthode de résolution et ses choix conceptuels et méthodologiques
- Degré de maîtrise
 - niveau de cohérence
 - liens logiques formant un ensemble
 - niveau de précision
 - clarté, concision
 - rigueur des concepts et techniques
 - niveau d'intégration
 - dans l'analyse
 - dans la recherche de solutions
 - niveau d'autonomie
 - sens de l'initiative
 - réflexion personnelle
 - respect du temps alloué



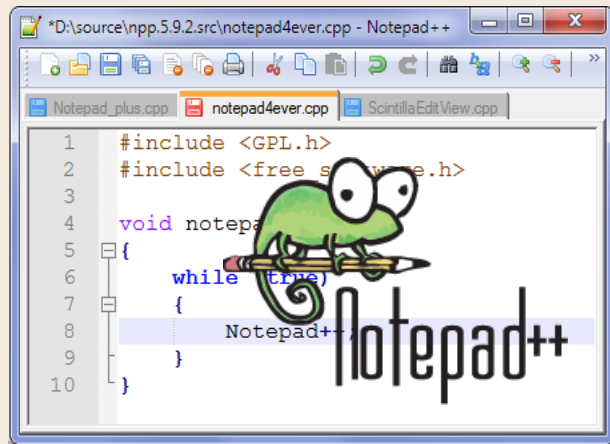
Ressources

- matière, exos, corrigés
 - GitHub
- administration, évaluation, communication
 - Moodle
 - 4IPO3 Programmation Orientée Objet

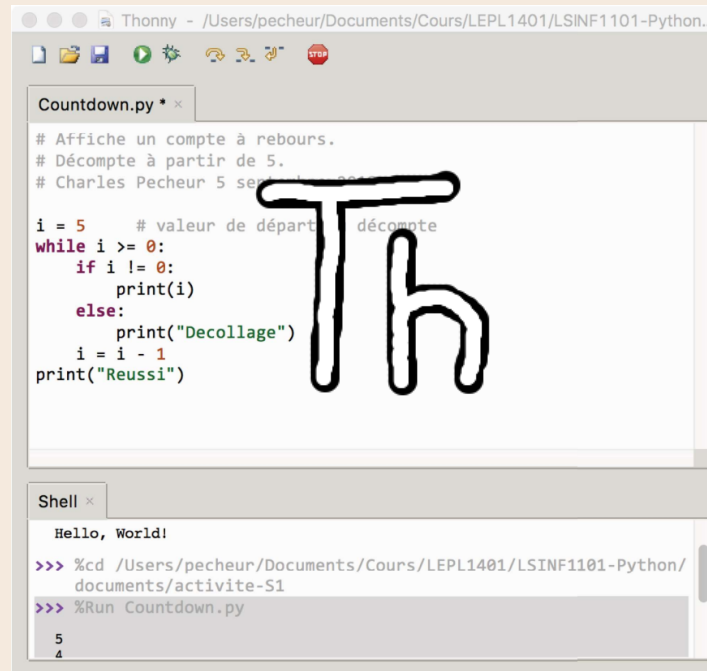


Outils : Editeur de texte vs. Integrated Development Environment

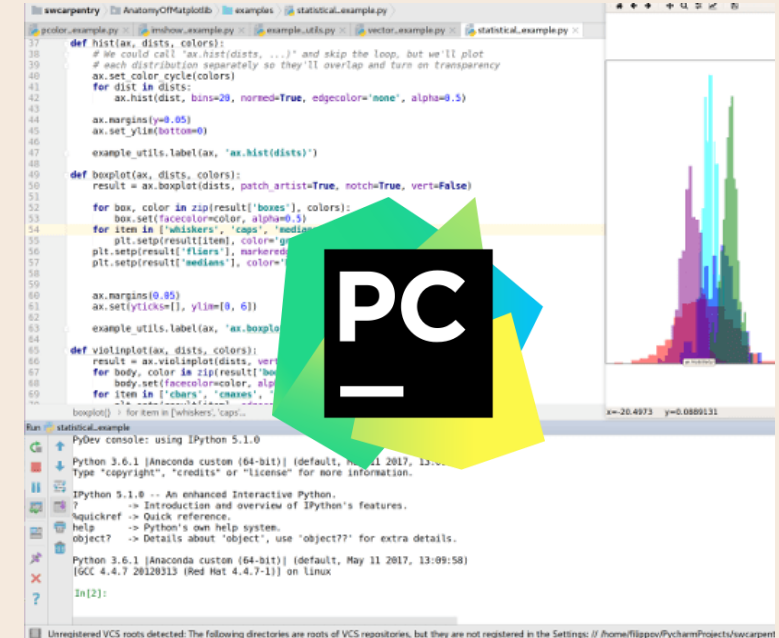
Notepad++ (éditeur)



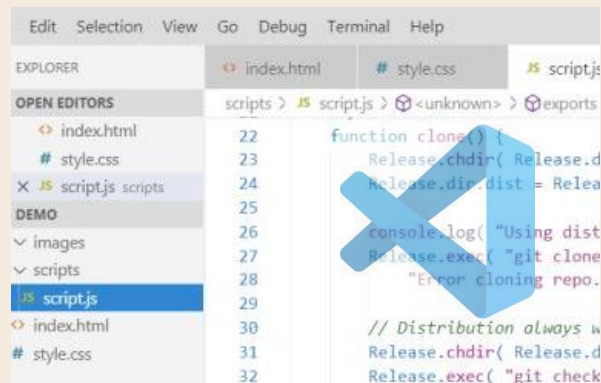
Thonny (IDE)



JetBrains PyCharm (IDE)



Visual Studio Code (IDE)



Outil : JetBrains PyCharm



- Accès à la version professionnelle JetBrains PyCharm
 - si inscription basée sur adresse e-mail *@isfce.be*
- Installation
 1. créer votre compte sur **jetbrains.com**
à l'aide de votre adresse e-mail *<nom>.<prenom>@isfce.be*
 2. télécharger et installer **PyCharm**
 3. s'enregistrer, option : *get license from JetBrains*

PS : procédure à confirmer

Table des matières

- Introduction
 - 00. Propos liminaires
 - 01. Programmation procédurale : rappel
- Concepts de Programmation Orientée Objet
 - 11. Programmation orientée objet : bases
 - 13. Programmation orientée objet : aspects avancés
 - 17. Patron de conception (design pattern)
- Applications de la POO
 - 21. Graphical User Interface



01. Programmation procédurale : rappel

- Langage Python
- Variable simple
- Variable complexe
- Instruction
- Fonction
- Structures de contrôle
- Listes

- Tuples
- Dictionnaires

 → Python : éléments

11. Programmation orientée objet : bases

- Module
- Manipulation d'un objet
- Conception et modélisation d'un objet
- Paramètre et méthode d'un objet
- Attribut public ou privé
- Encapsulation
- Variable et méthode de classe



→ Orienté Objet → 11-POO-basic



= BUROTIX ()

13. Programmation orientée objet : aspects avancés

- Associations entre objets
- Héritage simple
- Classe abstraite
- Polymorphisme
- Surcharge
- Héritage multiple

 → Orienté Objet → 13-POO-advanced

17. Patron de conception (*design pattern*)

- Decorator
- Factory
- Iterator
- Observer
- Singleton

 → Orienté Objet → 17-POO-design-pattern

Exos : les fils rouges

- Chaque exo porte un numéro $\langle SS \rangle - \langle CC \rangle - \langle T \rangle \langle N \rangle$
 - $\langle SS \rangle$: numéro de section
 - 11 : aspects élémentaires
 - 13 : aspects avancés
 - $\langle CC \rangle$: numéro du chapitre dans cette section
 - $\langle T \rangle$: thème de l'exo
 - 0 : drill
 - 1 : simulateur de trafic
 - 2 : jungle
 - 3 : gestion d'entreprise
 - $\langle N \rangle$: numéro de l'exo dans ce thème



PP vs. OOP

Programmation Procédurale	Programmation Orientée Objet
Données et fonctions séparées	Attributs et méthodes rassemblés autour d'un thème Frontière floue entre attribut et méthode
visions fonctionnelle et structurelle non liées	visions fonctionnelle et structurelle imbriquées
Risque de "code spaghetti"	Applications complexes aux évolutions incessantes
conception sans principe fédérateur	"Design pattern" (patron de conception)
Vision centralisée	Vision décentralisée
On gère un programme	On gère des classes



PP vs. OOP

Programmation Procédurale	Programmation Orientée Objet
Exécution chronologique Exécution "pas à pas"	Contraintes logiques entre classes Toutefois, exécution "pas à pas"
Programmation d'instructions	Modélisation et simulation du monde réel
Liaisons complexes entre fonctions	Liaisons simples entre classes
Risque de nombreux copiés-collés	Développement d'application itéré et évolutif Robustesse, stabilité, ré-emploi
Programmation éloignée de l'analyse	Approche abstraite possible, proche de l'analyse (UML)

Quelques langages OO

Java	C#	Python	C++	JavaScript	Node.js
compilé	compilé	interprété	compilé	interprété	comp. + inte.
full OO	full OO	procéd. + OO	procéd. + OO	prototype	prototype
générique	event-driven	générique	générique	event-driven	
héritage simple	héritage simple	héritage multiple	héritage multiple	prototype	prototype
tout est classe	tout est classe	tout est objet			
typage statique	typage statique	typage dynamique	typage statique	typage dynamique	typage dynamique

Ressources

- Bibliographie
 - Livre
 - "La programmation orientée objet"
 - Hugues Bersini
 - 7e édition, 2017
 - Cours UCL / EPL / LINFO1101
 - "Introduction à la programmation"
 - Kim Mens, Siegfried Nijssen, Charles Pecheur
 - Cours UCL / EPL / LSINF1225
 - "Conception orientée objet et gestion de données"

