

Journée des professeurs de SNT-NSI

4 Décembre 2024

Programmation Orientée Objet

Présentation de situations en classe

Échanges

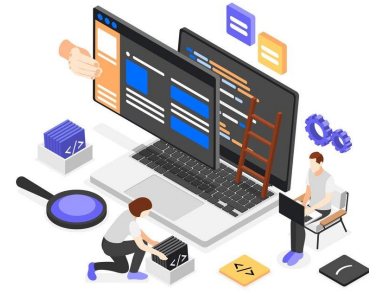


Issu du programme de NSI en Terminale

Contenus : Vocabulaire de la programmation objet : classes, attributs, méthodes, objets.

Capacités attendues : Écrire la définition d'une classe. Accéder aux attributs et méthodes d'une classe.

Commentaires : On n'aborde pas ici tous les aspects de la programmation objet comme le polymorphisme et l'héritage.



Programmation Orientée
Objet

Présentation de
situations en classe

Août - Rentrée : Fondamentaux 1ere

Septembre : POO

Projet « SpeedRun »

Projet « RPG » : Gestion Personnage

Octobre : Piles / Files

Projet « RPG » : Sauvegarde de donnée

Novembre : Graphes

Projet « RPG » : Compétences personnage

Bac Blanc NSI

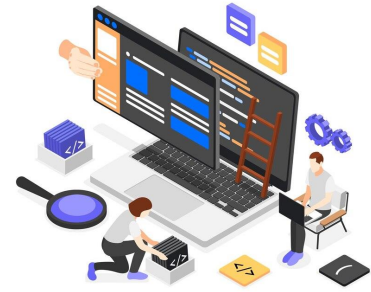
Décembre : Base de données

Janvier : Arbres

Exercices

Autres points :
récursivité...

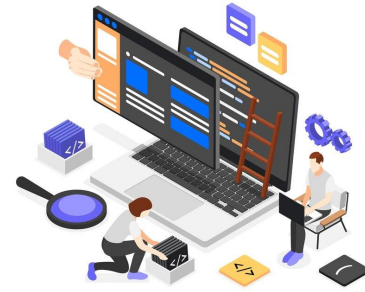
Devoirs



**Programmation Orientée
Objet**

Présentation de
situations en classe

Planification :



Programmation Orientée Objet

Présentation de
situations en classe

Capacités attendues :

Écrire la définition
d'une classe.

Sous forme de schéma

En langage Python

Structures de données		FICHE EXERCICES	
Programmation Orientée Objet (POO)			
ÉTABLISSEMENT	Boisjoly Potier	TERMINALE - NSI	ANNÉE SCOLAIRE 2024/2025

(1) Très insuffisant (2) Insuffisant (3) Satisfaisant (4) Très satisfaisant	1	2	3	4
Vocabulaire de la programmation objet : classes, attributs, méthodes, objets.				
Écrire la définition d'une classe. Accéder aux attributs et méthodes d'une classe.				

Exercice 3 : Thème Transports

Représenter sous forme de schéma une classe Vehicule avec :

- Les attributs suivants :
 - type_vehicule
 - nombre_de_places
 - vitesse_maximale
- Les méthodes suivantes :
 - __init__ (constructeur)
 - afficherVehicule (méthode)

Étapes :

- Définir la classe Vehicule avec un constructeur qui initialise les attributs type_vehicule, nombre_de_places, et vitesse_maximale.
- Créer une instance de la classe avec des valeurs de ton choix.
- Affiche les attributs de l'objet créé.

Sortie attendue :

```
Type de véhicule : Voiture  
Nombre de places : 5  
Vitesse maximale : 180 km/h
```

Didactique – Partie 2 – Mise en situation

Capacités attendues :

Accéder aux attributs et méthodes d'une classe.

Mini Projet « SpeedRun »



[Speed Run Vidéo](#)

Structures de données		FICHE TP	
Programmation objet			
ÉTABLISSEMENT	Boisjoly Potier	TERMINALE - NSI	ANNÉE SCOLAIRE 2024/2025

Compétences du BO NSI attendues :

(1) Très insuffisant (2) Insuffisant (3) Satisfaisant (4) Très satisfaisant	1	2	3	4
Vocabulaire de la programmation objet : classes, attributs, méthodes, objets.				
Écrire la définition d'une classe. Accéder aux attributs et méthodes d'une classe.				

Consignes :

Consigne pour la Réalisation du Jeu "Speed Run" avec Programmation Orientée Objet (POO).

Structures de données		FICHE TP	
Programmation objet			
ÉTABLISSEMENT	Boisjoly Potier	TERMINALE - NSI	ANNÉE SCOLAIRE 2024/2025

400x600 pixels et y intégrer un `canvas` pour afficher les éléments du jeu.
Structure du code :

- Une structure de base que vous pouvez suivre pour organiser votre code est fourni en fichier annexe `.py` :

Étapes de Réalisation :

Niveau 1 :

- Conception des Classes :
 - Classe `SpeedRunGame` : Cette classe gère l'ensemble du jeu. Elle s'occupe de l'initialisation de la fenêtre de jeu, du `canvas`, de la gestion du score, du démarrage et de la fin du jeu, ainsi que de la gestion des événements.
 - Classe `Car` : Cette classe représente la voiture contrôlée par le joueur. Elle gère le déplacement de la voiture en fonction des entrées utilisateur.
 - Classe `Obstacle` : Cette classe représente les obstacles (barrières) qui apparaissent sur la route. Elle gère leur création, leur déplacement, et la vérification des collisions avec la voiture.

Niveau 2 :

- Création et Gestion de la Voiture :
 - La classe `Car` doit charger une image de voiture et gérer son positionnement initial en bas de l'écran.
 - Implémentez des méthodes pour déplacer la voiture à gauche, à droite, en haut, et en bas, en réponse aux touches directionnelles du clavier.
- Création et Gestion des Obstacles :
 - La classe `Obstacle` doit charger une image de barrière et gérer leur apparition aléatoire en haut de l'écran.
 - Implémentez des méthodes pour déplacer les obstacles de haut en bas de l'écran.

Niveau 3 :

- Création et Gestion des Obstacles :
 - La classe `SpeedRunGame` doit gérer la création périodique des obstacles et vérifier les collisions avec la voiture.
- Gestion du Score et de la Fin de Jeu :
 - Le score doit être mis à jour chaque fois qu'un obstacle est évité avec succès.
 - Implémentez une méthode dans `SpeedRunGame` pour détecter la

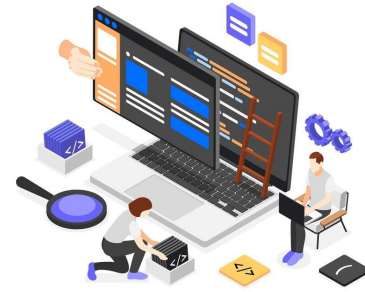
"Speed Run" en Python en utilisant la bibliothèque `Tkinter` pour éviter des obstacles (barrières) descendant de et d'éviter les obstacles et de cumuler des points entre en collision avec une barrière, et le

es `car.png` (voiture) et `barrier.png` (barrière) dans votre script Python. L'objet nécessite Python avec la librairie `Tkinter` professeur.

n projet en utilisant la `POO` en Python. L'intégration d'éléments graphiques avec `Tkinter` nécessite la gestion des événements et des

s pour organiser les fonctionnalités de votre jeu. : Assurez-vous que les événements clavier sont correctement gérés et que chaque classe ait une responsabilité claire et

e Graphique. : doit initialiser une fenêtre `Tkinter` de taille



Programmation Orientée Objet

Présentation de situations en classe

Didactique – Partie 3 – Consolidation par Répétition

Capacités attendues :

Écrire la
définition d'une
classe.

Accéder aux
attributs et
méthodes d'une
classe.



Répétitions

Structures de données	
Programmation Orientée	
ÉTABLISSEMENT	Boisjoly Potier

Exercice 4 : Gestion de

Schéma :

Un véhicule a une vitesse et une

Classe : `Vehicule`

Attributs : `vitesse (privé)`

Méthodes : `get_vitesse`, `acc`

Énoncé :

Créez une classe `Vehicule` avec :

- Un attribut privé `_vitesse`
- Une méthode `get_vitesse`
- Une méthode `accelerer`

Simulez deux accélérations pour

Exemple :

```
class Vehicule:
    def __init__(self):
        self._vitesse = 0

    def get_vitesse(self):
        return self._vitesse

    def accelerer(self):
        self._vitesse += 10
```

```
# Exemple d'utilisation
voiture = Vehicule()
voiture.accelerer()
voiture.accelerer()
print(voiture.get_vitesse()) # 20
```

Structures de données		FICHE EXERCICES	
Programmation Orientée Objet (POO)			
ÉTABLISSEMENT	Boisjoly Potier	TERMINALE - NSI	ANNÉE SCOLAIRE 2024/2025

Exercice 5 : Gestion des comptes bancaires (Méthodes spéciales)

Schéma :

Un compte bancaire possède un solde et permet des dépôts et des retraits.

Classe : `CompteBancaire`

Attributs : `solde`

Méthodes : `deposer`, `retirer`, `__str__`

Énoncé :

Créez une classe `CompteBancaire` avec :

- Un attribut `solde` initialisé à 0.
- Une méthode `deposer` pour ajouter une somme au solde.
- Une méthode `retirer` pour retirer une somme (vérifiez que le solde reste positif).
- Une méthode spéciale `__str__` pour afficher le solde actuel sous forme de chaîne.

Testez en déposant 200€, retirant 50€, puis affichez le solde.

Exemple :

```
class CompteBancaire:
    def __init__(self):
        self.solde = 0

    def deposer(self, montant):
        self.solde += montant

    def retirer(self, montant):
        if montant <= self.solde:
            self.solde -= montant
        else:
            print("Fonds insuffisants.")

    def __str__(self):
        return f"Solde actuel: {self.solde}€"

# Exemple d'utilisation
compte = CompteBancaire()
compte.deposer(200)
compte.retirer(50)
print(compte) # "Solde actuel: 150€"
```



Programmation Orientée
Objet

Présentation de
situations en classe

Évaluation des acquis

BACCALAURÉAT
SESSION 2025
Épreuve de l'enseignement de spécialité
NUMÉRIQUE et SCIENCES INFORMATIQUES
Partie pratique « blanche »
 Lycée BoisjolyPotier
 Classe Terminale de la voie générale

Sujet n°2

DURÉE DE L'ÉPREUVE : 1h30

Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3

Dès que le sujet vous est remis, assurez-vous qu'il est complet.
 Le candidat doit traiter tous les exercices.

NSI_BAP_2024_03_21_14_09

		1	2	3	4		
Modalités EP Blanche (Auto-évaluation)	(1) Très insuffisant (2) Insuffisant (3) Satisfaisant (4) Très satisfaisant						cartes à
	La partie pratique comportera 4 exercices (2 exercices de programmations, 2 exercices permettant de valider les compétences de programmations)						is données dans les
Modalités (Auto-évaluation)	La partie pratique consiste en la résolution de deux exercices sur ordinateur, chacun étant noté sur 10 points. Le candidat est évalué sur la base d'un dialogue avec un professeur-examinateur.						la condition est vraie. Si
	Le premier exercice consiste à programmer un algorithme figurant explicitement au programme, ne présentant pas de difficulté particulière. Pour le second exercice, un programme est fourni au candidat. Cet exercice ne demande pas l'écriture complète d'un programme, mais permet de valider des compétences de programmation.						

```

""" Les cartes à jouer ont une couleur et un numéro. """
valeurs = ["As", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Valet", "Dame", "Roi"]
return valeurs[seuil-1]

def recuperer_couleur(seuil):
    """ Récupère la couleur de la carte.
    paramètre : seuil, couleur, carreau, trèfle. """
    couleurs = ["pique", "coeur", "carreau", "trèfle"]
    return couleurs[seuil-1]

class Paquet_de_cartes:
    def __init__(self):
        """ Initialise l'attribut contenu avec une liste des 52
        objets Carte possédant les attributs par valeurs constantes en
        commençant par pique, puis cœur, carreau et trèfle. """
        ...

def recuperer_carte(seuil, pos):
    """ Récupère la carte qui se trouve à la position pos
    (entière comprise entre 0 et 51). """
    ...

Exemple :
>>> jeu = Paquet_de_cartes()
>>> carte1 = jeu.recuperer_carte(20)
>>> carte1.recuperer_valeur() \
+ de " + carte1.recuperer_couleur()
"8 de coeur"
>>> carte2 = jeu.recuperer_carte(0)
>>> carte2.recuperer_valeur() \
+ de " + carte2.recuperer_couleur()
"As de pique"
>>> carte3 = jeu.recuperer_carte(52)
AttributeError : paramètre pas valide
    
```

Écrite

Pratique

Analyses des attendus et des difficultés rencontrés

Structures de données	ÉVALUATION		
Programmation Orientée Objet (POO)			
ÉTABLISSEMENT Boisjoly Potier	TERMINALE 1SG	ANNÉE SCOLAIRE	2024/2025

Question 3 (3 points) :
 Reprenez le thème Gestion de produits en magasin, créez une classe `ProduitAlimentaire` héritant de la classe `Produit`. Ajoutez un attribut supplémentaire `bio` (booléen) pour indiquer si le produit est biologique.

- Instanciez un objet de cette classe pour un produit "Pomme", prix 1.5, avec `bio=True`
- Affichez ses informations complètes (nom, prix, bio).

Question 4 (3 points) :
 Reprenez le thème Gestion de véhicules, modifiez la classe `Vehicule` pour ajouter une méthode `ralentir` qui réduit la vitesse de 5 km/h (sans descendre en dessous de 0). Simulez une accélération de 20 km/h, puis deux ralentissements. Affichez la vitesse à chaque étape.

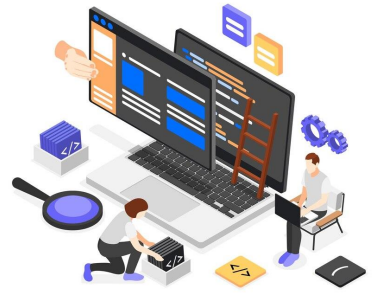
Structures de données	ÉVALUATION		
Programmation Orientée Objet (POO)			
ÉTABLISSEMENT Boisjoly Potier	TERMINALE 1SG	ANNÉE SCOLAIRE	2024/2025

Question 9 (2 points) :
 Reprenez le thème Gestion des étudiants, ajoutez une méthode `changer_nom` dans la classe `Etudiant` pour modifier dynamiquement le nom d'un étudiant.

- Instanciez un étudiant avec le nom "MARLIN", puis modifiez son nom en "Dourand".
- Affichez l'ancien et le nouveau nom.

Question 10 (2 points) :
 Définissez une classe abstraite `Forme` avec une méthode abstraite `calculer_surface`. Créez deux classes filles `Rectangle` et `Cercle` qui implémentent cette méthode.

- Testez en créant un rectangle de dimensions 5x3 et un cercle de rayon 2.
- Affichez leurs surfaces respectives.



Programmation Orientée Objet

Présentation de situations en classe

Projet transverse : Jeu type « RPG »

Objectif pédagogique

Utilisation des acquis dans un projet « type entreprise »

Travail en groupe de 4/5 élèves

Mise en autonomie



[Speed Run Vidéo](#)



**Programmation Orientée
Objet**

Présentation de
situations en classe