
Examen POO Python

EPSI - SN2 2022

Geoffroy Ladrat

2022 - 2023




Table des matières

Examen Python POO - SN2	2
Barème et notation	2
Instructions de l'examen	2
Format du livrable rendu en fin de session	2
Partie 1 : Le prototype Robot (5pt)	3
Créez une classe Robot	3
Créez son constructeur (méthode <code>__init__()</code>)	3
Rendez lisible l'affichage d'un robot par un humain (méthode <code>__str__()</code>)	3
Déposez un brevet sur votre technologie	3
Partie 2 : Production d'une source d'énergie, le Réacteur Arc (9pt)	4
Créez une classe ReacteurArc	4
Créez son constructeur (méthode <code>__init__()</code>)	4
Alimentez le noyau pour allumer le Réacteur Arc	4
Intégration du Réacteur Arc aux Robots	4
Configurer le démarrage du robot	5
Donnez vie à votre premier robot	5
Partie 3 : Sécurisation du Robot (6pt)	6
Créez une classe RobotSecurise	6
Générez un identifiant sécurisé	6
Modifier l'affichage du robot	6
En route vers le futur...	6

Examen Python POO - SN2

Barème et notation

La note finale du module Python - POO (Programmation Orientée Objet) est une note sur 20, le barème est indiqué à chaque partie.

La notation repose sur l'analyse de :

- **Compréhension de l'énoncé**
- **Respect des conventions** et des structures de mot-clefs
- **Résultat de l'algorithme** (bug, respect de la demande, etc.)

Instructions de l'examen

Autorisé :

- Utiliser le projet `dice_warriors` réalisé lors des précédents cours
- Effectuer des recherches sur Google

Interdit :

- **Discord** ou autre outil du type est strictement interdit et peut mener à une exclusion de l'examen
- **La copie de toute ou partie du code** d'un autre étudiant est sanctionnée d'une perte de point pour tous les concernés

En cas de doute sur l'autorisation ou l'interdiction de quelque chose, demander au formateur.

Format du livrable rendu en fin de session

Vous devez rendre un fichier `robot.py` contenant le code et les commentaires demandés dans le sujet. Vous pouvez quitter l'examen une fois le livrable reçu par e-mail et validé par le formateur.

- Vous pouvez compresser le fichier dans une archive `.zip, .rar, etc.`
- Si vous oubliez le fichier joint et quittez l'examen, vous risquez de perdre de nombreux points, soyez vigilant !

Partie 1 : Le prototype **Robot** (5pt)

Ingénieur renommé, vous avez fondé il y a quelques temps l'entreprise **RoboTech** et vous travaillez nuit et jour sur un concept de robot intelligent. Vous êtes à quelques pas de faire prendre vie à votre création...

Créez une classe **Robot**

Elle dispose de l'attribut de classe :

- `_type` : Initialisé à `"Robot"`

Créez son constructeur (méthode `__init__()`)

Il prend les arguments suivants :

- `prenom` (par défaut à la valeur `John`)
- `nom` (par défaut à la valeur `Doe`)

Il initialise les attributs suivants :

- `_prenom` : Valeur de l'argument correspondant
- `_nom` : Valeur de l'argument correspondant

Rendez lisible l'affichage d'un robot par un humain (méthode `__str__()`)

Appliquez les concepts vus en cours sur cette méthode. Vous pouvez afficher ce que vous voulez, mais elle doit afficher au moins son attribut `_type` et son `prenom` .

Déposez un brevet sur votre technologie

Afin d'éviter les copies des concurrents, vous devez déposer un brevet sur votre technologie. L'autorité compétente vous demande des informations pour cela :

Dans un commentaire dans votre code, expliquez en quelques mots simples :

- Le concept de `constructeur` en POO (Programmation Orientée Objet)
- La différence entre une `classe` et une `instance` d'un objet

Partie 2 : Production d'une source d'énergie, le Réacteur Arc (9pt)

Afin de pouvoir rendre autonome le fonctionnement de votre création, vous avez longtemps cherché une source d'énergie suffisante. Vous et votre équipe finalisez enfin la première version du système d'énergie du robot : le Réacteur Arc .

Créez une classe ReacteurArc

Créez la classe demandée.

Créez son constructeur (méthode `__init__()`)

Il initialise l'attribut suivant :

- `_noyau` : Valeur par défaut `False`

Alimentez le noyau pour allumer le Réacteur Arc

Créez une méthode `allumer_noyau(source)` .

Elle possède donc un argument `source` et fonctionne comme suit :

- Si le paramètre `source` est égal à `"Palladium"` , l'attribut `_noyau` passe à `True`
 - Affiche le message `"Alimentation en Palladium - ok"`
 - Affiche le message `"Démarrage du Réacteur Arc - ok"`
- Sinon il ne se passe rien

Intégration du Réacteur Arc aux Robots

Dans le constructeur de la classe `Robot` :

- Créez un nouvel attribut `_reacteur_arc`
- Instanciez un nouvel objet `ReacteurArc` dans cet attribut

Configurer le démarrage du robot

Dans la classe `Robot`, créez une méthode `allumage_robot()`, qui a pour effet :

- Affiche `"Tentative de démarrage du robot..."`
- Appelle la méthode `allumer_noyau("Palladium")` de son attribut `_reacteur_arc`
- Si le réacteur a correctement démarré → Affiche le message de votre choix pour prouver que vous avez créé le premier robot intelligent de l'humanité !

Donnez vie à votre premier robot

Dans un bloc d'instructions utilisant la super-variable `__name__` :

- Créez un robot avec le prénom et le nom de votre choix
 - Affichez le robot pour présenter votre création à vos investisseurs
 - Démarrez le robot
-

Partie 3 : Sécurisation du **Robot** (6pt)

Votre invention commence à attirer les regards et les jalousies. Après plusieurs tentatives de vol de votre technologie, vous décidez de sécuriser votre prototype en lui attribuant un identifiant qui permet de limiter son contrôle.

Créez une classe **RobotSecurise**

Cette classe hérite de la classe **Robot** et modifie son attribut de classe `_type` (valeur : `"Robot sécurisé"`). Dans le constructeur, ajoutez l'initialisation des attributs suivants :

- `_identifiant` : Valeur par défaut `""`

Générez un identifiant sécurisé

Créez une méthode `generer_identifiant()`.

Elle affecte la combinaison de chaîne de caractères suivante à l'attribut `_identifiant` :

- `"#robot.prenom_nom#" (où prenom et nom sont les attributs correspondant du robot)`

Modifier l'affichage du robot

Modifiez la méthode `__str__()` pour :

- Afficher l'identifiant du Robot, uniquement si son Réacteur Arc est démarré.
- Sinon affichez l'ancienne méthode `__str__()` de la classe **Robot**

En route vers le futur...

Créez un robot sécurisé, générez son identifiant, démarrez-le, présentez-le à l'humanité.

Le rendu final du terminal pourra par exemple être le suivant :

```
Tentative de démarrage du robot...
Alimentation en Palladium - ok
Démarrage du réacteur Arc - ok
I'm alive !
Robot sécurisé #robot.Astro_Boy# - Hello World !
```