

NOM Prénom :

CONTROLE CONTINU – MEMOIRE ET LOGICIEL : TECHNIQUE DE PROGRAMMATION 2

- Temps de travail : **2h**
- Première heure : documents non-autorisés
- Deuxième heure : documents et accès internet autorisés

Attention : si vous utilisez les propos ou idées d'autrui sans créditer l'auteur, y compris ChatGPT, vous serez sanctionné. La **version payante** de ChatGPT ou autre bot est interdite d'usage.

La **présentation** (lisibilité, orthographe) et le **respect des consignes** suivantes seront pris en compte dans la note finale.

PREMIERE HEURE : DOCUMENTS NON-AUTORISES

Notions générales

Exercice 1

/6

Quelle est la différence entre une classe et un objet en Python ? Donnez un exemple concret.

Classe = définition d'un type d'objet | Objet = un exemple spécifique d'une classe / d'un type d'objet. Par exemple une bibliothèque est une classe, qui a des attributs et des fonctions spécifiques à cette classe mais partagés entre les objets de même type (classe). Et la bibliothèque nationale est un objet qui a des attributs spécifiques à lui-même comme le nom de la bibliothèque, l'adresse, etc.

En quoi consiste le mécanisme d'héritage en programmation orientée objet ? Donnez un exemple (autre que ceux du cours) d'héritage simple en Python.

Création de relation entre classes sous forme d'une arborescence du plus général au plus spécifique. Une classe peut ainsi hériter d'attributs et méthodes d'une classe mère.

Exemple : Œuvre, Film(Œuvre), LongMetrage(Film).

(explication de relation entre ces classes)

Qu'est-ce qu'un assesseur (getter) et un mutateur (setter) dans le contexte de la programmation orientée objet ? Donnez un exemple en Python.

Un accesseur est une méthode qui permet d'afficher un attribut qui est privé et un

NOM Prénom :

mutateur est une **méthode** qui permet de **modifier** un **attribut privé**. **Exemple** :

self.__name est un attribut privé. **getName()** est un getter qui retourne **self.__name**. Et

setName(new_name) est un mutateur qui instancie la variable **self.__name** à

new_name.

Exemple de classe et objet

Exercice 2

Voici un extrait de code python :

/5

Classe

```
class Animal:
    def __init__(self, name, species):
        self.name = name
        self.species = species
        self.energy = 100 # énergie de l'animal

    def make_sound(self):
        """ simule le fait que l'animal émet un son."""
        if self.energy >= 10:
            print(f"{self.name} fait un bruit caractéristique!")
            self.energy -= 10
        else:
            print(f"{self.name} est trop fatigué pour faire du bruit.")

mon_animal = Animal(name="Rex", species="Chien")
mon_animal.make_sound()
```

Méthode spéciale : constructeur

Méthode

Dans le code ci-dessus identifiez :

- classe(s)
- méthode(s)
- attribut(s) : **name, species, energy**
- instance(s)

Qu'est censé afficher ce code ?

Rex fait un bruit caractéristique!

.....

.....

NOM Prénom :

DEUXIEME HEURE : DOCUMENTS ET ACCES INTERNET AUTORISES

→ CORRECTIONS DANS CC_TESTCORRECTION.IPYNB

Téléchargez et ouvrez le notebook « **cc_test.ipynb** » disponible sur Moodle ([Contrôle continu 1](#)) et renommez-le comme suit :

- nomPrenom_cc_test.ipynb

Attention : choisissez bien le bon fichier !

Une fois les exercices terminés, déposez le notebook sur moodle :

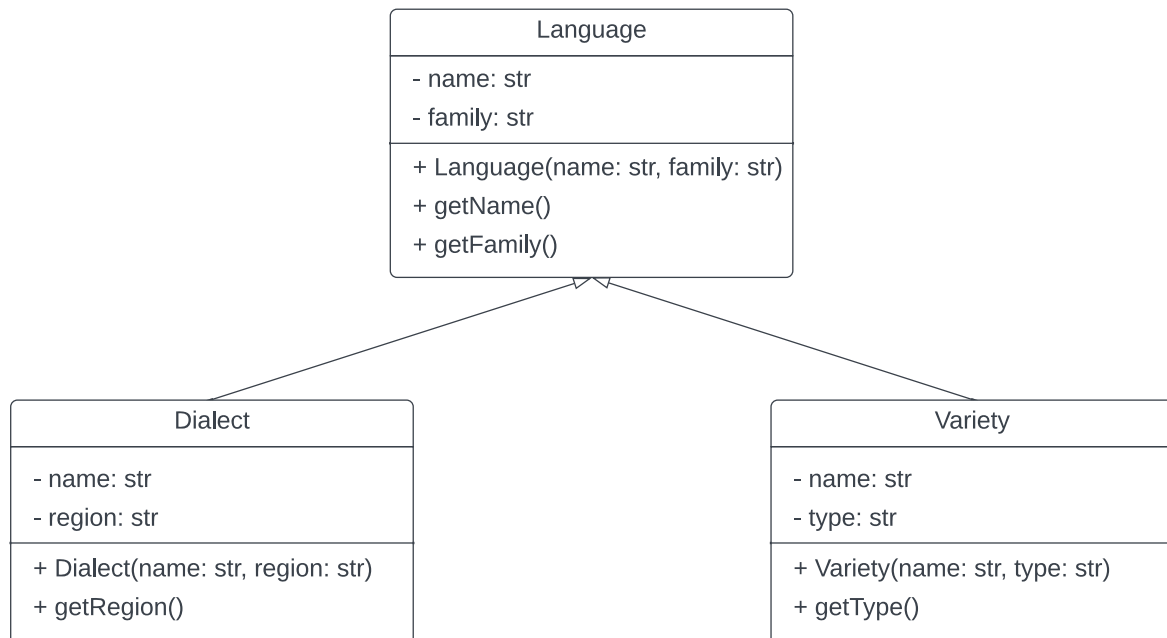
- [Contrôle continu 1](#)

Héritage

Exercice 3

Ecrivez un code python à partir de cet UML. Rappel : commentez-bien votre code.

/3



Exception et erreurs

Exercice 4

Dans le notebook téléchargé, vous trouverez un extrait de code rempli d'erreurs.

- Identifiez les types d'erreur
- Mettez la ligne en commentaire pour garder une trace de l'erreur
- Et proposez une correction.

Remarque : cet exercice a été testé avec un chatbot, celui-ci ne retrouve pas ni ne renvoie les bonnes réponses.

/5

NOM Prénom :

CONTROLE CONTINU – MEMOIRE ET LOGICIEL : TECHNIQUE DE PROGRAMMATION 2

- Temps de travail : **2h**
- Première heure : documents non-autorisés
- Deuxième heure : documents et accès internet autorisés

Attention : si vous utilisez les propos ou idées d'autrui sans créditer l'auteur, y compris ChatGPT, vous serez sanctionné. La **version payante** de ChatGPT ou autre bot est interdite d'usage.

La **présentation** (lisibilité, orthographe) et le **respect des consignes** suivantes seront pris en compte dans la note finale.

PREMIERE HEURE : DOCUMENTS NON-AUTORISES

Notions générales

Exercice 1

Expliquez brièvement ce qu'est une classe en programmation orientée objet et comment elle est utilisée en Python.

/6

Une classe est un modèle ou un plan pour créer des objets. Elle définit les propriétés (attributs) et les comportements (méthodes) communs à un ensemble d'objets. En Python, une classe est définie à l'aide du mot-clé `class`, suivi du nom de la classe et d'un bloc d'attributs et de méthodes. Pour utiliser une classe, il faut créer une instance de cette classe en appelant son constructeur.

Définissez le terme "méthode" en programmation orientée objet. Fournissez un exemple de méthode dans une classe Python.

Une méthode est une fonction définie à l'intérieur d'une classe qui agit sur les données de cette classe. Les méthodes sont utilisées pour définir les comportements ou les actions qu'un objet peut effectuer. Exemple : `__str__()` est une méthode de classe en python

.....

.....

Qu'est-ce que l'abstraction en programmation orientée objet ? Donnez un exemple (autre que ceux du cours) en Python.

NOM Prénom :

L'abstraction fait référence au concept de **masquer les détails internes** d'un objet.

Cela permet de **simplifier** l'utilisation de l'objet en cachant les complexités de son implémentation. Exemple : la classe Telephone définit l'interface **publique** d'un objet, en spécifiant ses méthodes (appel) et ses attributs (marque), tandis que les détails de **l'implémentation des sous-classes** Portable et Fixe sont cachés à l'utilisateur.

Exemple de classe et objet

Exercice 2

Voici un extrait de code python :

/5

Classe

```
class Restaurant:
    def init (self, name, cuisine_type):
        self.name = name
        self.cuisine_type = cuisine_type
        self.is_open = False # le restaurant est ouvert ou pas

    def open_restaurant(self):
        """simule l'ouverture du restaurant."""
        self.is_open = True
        print(f"{self.name} est maintenant ouvert ! Bienvenue !")

mon_restaurant = Restaurant(name="Le Délice", cuisine_type="Française")
mon_restaurant.open_restaurant()
```

Méthode spéciale :
constructeur

Méthode

Dans le code ci-dessus identifiez :

- classe(s)
- méthode(s)
- attribut(s) : name, cuisine_type, is_open
- instance(s)

Qu'est censé afficher ce code ?

Le Délice est maintenant ouvert ! Bienvenue !

.....

.....

NOM Prénom :

DEUXIEME HEURE : DOCUMENTS ET ACCES INTERNET AUTORISES

→ CORRECTIONS DANS CC_TEST2CORRECTION.IPYNB

Téléchargez et ouvrez le notebook « **cc_test2.ipynb** » disponible sur Moodle ([Contrôle continu 1](#)) et renommez-le comme suit :

- nomPrenom_cc_test2.ipynb

Attention : choisissez bien le bon fichier !

Déposez le notebook sur moodle :

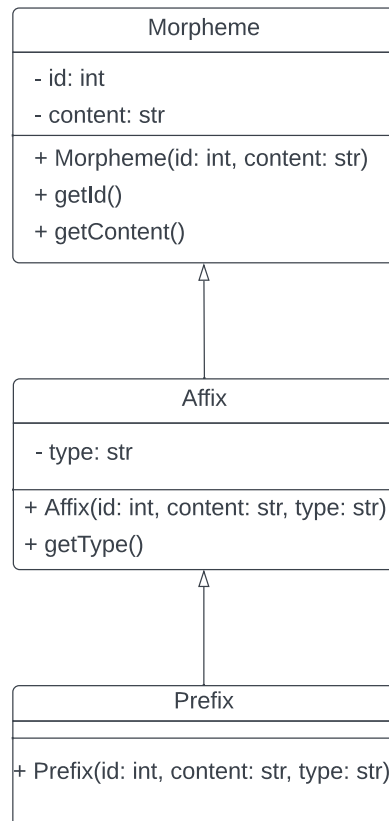
- [Contrôle continu 1](#)

Héritage

Exercice 3

Ecrivez un code python à partir de cet UML. Rappel : commentez-bien votre code.

/3



Exception et erreurs

Exercice 4

Dans le notebook téléchargé, vous trouverez un extrait de code rempli d'erreurs.

- Identifiez les types d'erreur
- Mettez la ligne en commentaire pour garder une trace de l'erreur
- Et proposez une correction.

Remarque : cet exercice a été testé avec un chatbot, celui-ci ne retrouve pas ni ne renvoie les bonnes réponses.

/5