

Python — Programmation orientée objet et récursivité

NSI · Première
Contrôle n°1 · 60 min · sur 20

Nom et prénom

Classe

Date

NOTE

... / 20

CONSIGNES

- Durée : **60 minutes**. Le temps conseillé par exercice est indiqué à côté du barème.
- Le barème est donné : il dit où passer du temps.
- Écris tes réponses sur les lignes prévues. Une réponse hors des lignes reste comptée.

AVANT DE COMMENCER — CE QU'IL FAUT SAVOIR

- Objet modifiable : se demander si l'on veut un alias ou une copie.
- L'héritage se vérifie par le test « est un ».
- Une fonction récursive sans cas de base sature la pile d'appels.
- Correction et efficacité sont deux propriétés distinctes.
- Un programme qui masque ses erreurs échoue silencieusement.
- L'encapsulation sert la cohérence de l'objet, non la dissimulation.

Exercice 1 — Noms et objets

4 pts 11 min

Distinguer alias et copie

Réponds à chaque question.

1. Que fait une affectation de variable ?

1 pt

2. Pour quels types la question ne se pose-t-elle pas, et pourquoi ?

1 pt

3. Dans quels trois cas la question se pose-t-elle pour un objet modifiable ?

1 pt

4. Comment obtenir une vraie copie indépendante ?

1 pt

Exercice 2 — Classes et héritage

5 pts 14 min

Analyser la programmation orientée objet

Réponds à chaque question.

1. Qu'est-ce qu'une classe, et qu'est-ce qu'une instance ? 1 pt

2. Que partagent et que ne partagent pas deux objets d'une même classe ? 1 pt

3. Qu'est-ce que l'encapsulation ? 1 pt

4. À quoi sert-elle réellement ? 1 pt

5. Quel test permet de décider s'il faut employer l'héritage ? 1 pt

Exercice 3 — La récursivité

4 pts 11 min

Écrire et analyser une fonction récursive

Réponds à chaque question.

1. Quels deux éléments une fonction récursive exige-t-elle ? 1 pt

2. Que se passe-t-il sans progression stricte vers le cas de base ? 1 pt

3. Écris en français le cas de base et le cas général de la factorielle. 1 pt

4. Pourquoi une récursivité naïve sur Fibonacci est-elle correcte et inutilisable ? 1 pt

Exercice 4 — Exceptions et modules

4 pts 13 min

Employer exceptions et modules

Réponds à chaque question.

1. Que signale une exception ? 1 pt

2. Quelles sont les deux règles d'usage du bloc try/except ? 1 pt

3. Pourquoi un échec silencieux est-il pire qu'un arrêt net ? 1 pt

4. Pourquoi importer un module existant plutôt que réécrire une fonction ? 1 pt

Exercice 5 — Trouve l'erreur

3 pts 11 min

Repérer une erreur et la corriger

Chaque affirmation est fausse. Écris ce qu'il fallait dire.

1. « J'ai affecté ma liste à une nouvelle variable : j'ai donc une copie indépendante. » 1 pt

2. « Ma classe Voiture hérite de la classe Moteur, puisqu'une voiture a un moteur. » 1 pt

3. « Je rattrape toutes les exceptions pour que mon programme ne s'arrête jamais. » 1 pt