

Programmation orientée objet : Classes et objets - Quiz

<https://github.com/heig-vd-progim-course/heig-vd-progim2-course>

Visualiser le contenu complet sur GitHub [à cette adresse](#).

V. Guidoux, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Plus de détails sur GitHub

Cette présentation est un quiz pour tester ses connaissances sur le chapitre en cours. Pour plus de détails, consultez le [contenu complet sur GitHub](#).

Question 1 - Donnée

Complétion : Déclaration d'une classe

Quel code déclare correctement une classe `Book` en Java ?

- **A.** `class Book { }`
- **B.** `Book class { }`
- **C.** `class book { }`
- **D.** `new Book { }`

Question 1 - Réponse

Réponse correcte : A `class Book { }`

Explications :

- **A** : syntaxe correcte, le nom de la classe commence par une majuscule
- **B** : le mot-clé `class` doit être avant le nom
- **C** : par convention, les noms de classes commencent par une majuscule
- **D** : `new` sert à créer des objets, pas à déclarer des classes

Question 2 - Donnée

Complétion : Déclaration d'attributs

Quel code déclare correctement un attribut `title` de type `String` dans une classe ?

- **A.** `String title;`
- **B.** `title String;`
- **C.** `String = title;`
- **D.** `var title: String;`

Question 2 - Réponse

Réponse correcte : **A** `String title;`

```
class Book {  
    String title; // Déclaration correcte  
}
```

Explications :

- **B** : l'ordre est inversé
- **C** : utilise l'opérateur d'affectation au lieu de la déclaration
- **D** : syntaxe d'autres langages (TypeScript, Kotlin), pas Java

Question 3 - Donnée

Complétion : Constructeur manquant

Complétez le constructeur de la classe `Car` :

```
class Car {
    String brand;
    int year;

    -----(String brand, int year) {
        this.brand = brand;
        this.year = year;
    }
}
```

Question 3 - Réponse

Réponse correcte : `public Car(String brand, int year)`

```
public Car(String brand, int year) {  
    this.brand = brand;  
    this.year = year;  
}
```

Explications :

- Un constructeur a le **même nom que la classe**
- Un constructeur **n'a pas de type de retour** (même pas `void`)

Question 4 - Donnée

Complétion : Utilisation du mot-clé `this`

Pourquoi utilise-t-on `this` dans ce code ?

```
public Person(String name, int age) {  
    this.name = name;  
    this.age = age;  
}
```

Question 4 - Réponse

Réponse correcte : B - `this` fait référence à l'instance courante et différencie les attributs des paramètres

```
public Person(String name, int age) {  
    this.name = name; // this.name = attribut, name = paramètre  
    this.age = age;  
}
```

Explications :

- Sans `this`, on modifierait le paramètre, pas l'attribut

Question 5 - Donnée

Prédiction : Création d'objet

Combien d'objets `Product` sont créés dans ce code ?

```
Product laptop = new Product("Laptop", 899.90);  
Product mouse = new Product("Mouse", 25.50);  
Product screen = laptop;
```

A. 1 objet **B.** 2 objets **C.** 3 objets **D.** 4 objets

Question 5 - Réponse

Réponse correcte : B - 2 objets sont créés

```
Product laptop = new Product("Laptop", 899.90); // Objet 1
Product mouse = new Product("Mouse", 25.50); // Objet 2
Product screen = laptop; // Pas de nouvel objet, screen pointe vers laptop
```

Explications :

- `new` crée un nouvel objet en mémoire
- `screen = laptop;` copie la référence, pas l'objet
- `screen` et `laptop` pointent vers le **même objet**

Question 6 - Donnée

Prédiction : Indépendance des objets

Que vaut `book2.price` après l'exécution de ce code ?

```
Book book1 = new Book("Java Guide", 45.0);  
Book book2 = new Book("Python Basics", 35.0);  
book1.price = 50.0;
```

A. 35.0 **B.** 45.0 **C.** 50.0 **D.** Erreur de compilation

Question 6 - Réponse

Réponse correcte : A - `book2.price` vaut `35.0`

```
Book book1 = new Book("Java Guide", 45.0);  
Book book2 = new Book("Python Basics", 35.0);  
book1.price = 50.0; // Modifie uniquement book1
```

Explications :

- Chaque objet est **indépendant** et a ses propres valeurs
- Modifier `book1.price` n'affecte **pas** `book2.price`
- `book2` conserve sa valeur initiale de `35.0`

Question 7 - Donnée

Comparaison : Classe vs Objet

Quelle affirmation est correcte ?

- A.** Une classe est une instance d'un objet
- B.** Un objet est un modèle pour créer des classes
- C.** Une classe est un modèle pour créer des objets
- D.** Les termes "classe" et "objet" sont synonymes

Question 7 - Réponse

Réponse correcte : C - Une classe est un modèle pour créer des objets

Analogie :

- **Classe** = plan d'architecte pour une maison
- **Objet** = maison réelle construite à partir du plan

Différences :

- **Classe** : définit les attributs et méthodes (modèle abstrait)
- **Objet** : instance concrète avec des valeurs spécifiques

Question 9 - Donnée

Comparaison : Attribut vs Variable locale

Quelle est la différence entre ces deux variables `name` ?

```
class Person {  
    String name; // Variable 1  
  
    void displayInfo() {  
        String name = "Temporary"; // Variable 2  
        System.out.println(name);  
    }  
}
```

Question 9 - Réponse

Variable 1 = attribut, Variable 2 = variable locale

```
class Person {
    String name; // ATTRIBUT : appartient à l'objet

    void displayInfo() {
        String name = "Temporary"; // VARIABLE LOCALE : dans la méthode
        System.out.println(name); // Affiche "Temporary"
        System.out.println(this.name); // Affiche l'attribut
    }
}
```

Question 10 - Donnée

Comparaison : Méthode void vs avec retour

Quelle est la différence entre ces deux méthodes ?

```
// Méthode A
void displayAge() {
    System.out.println("Âge: " + age);
}

// Méthode B
int getAge() {
    return age;
}
```

Question 10 - Réponse

A affiche directement, B retourne une valeur

```
/**
 * Affiche l'âge directement
 */
void displayAge() {
    System.out.println("Âge: " + age);
}

/**
 * Retourne l'âge pour utilisation ultérieure
 */
int getAge() {
    return age;
}
```

Utilisation :

```
// Affiche directement
person.displayAge();
// Récupère la valeur
int age = person.getAge();
// Utilise dans une condition
if (person.getAge() > 18) { }
```

Question 12 - Donnée

Modification : Correction d'erreur

Ce code ne compile pas. Quelle est l'erreur ?

```
class Book {  
    String title;  
    double price;  
  
    void Book(String title, double price) {  
        this.title = title;  
        this.price = price;  
    }  
}
```

Question 12 - Réponse

Réponse correcte : Un constructeur ne doit pas avoir de type de retour

```
// ERREUR : un constructeur ne doit pas avoir de type de retour, même pas void
void Book(String title, double price) { // ERREUR : void

// ---

// CORRECT : pas de type de retour
public Book(String title, double price) { // Pas de void
    this.title = title;
    this.price = price;
}
```

Questions

Est-ce que vous avez des questions sur le quiz ou le contenu ?

Sources

- [Illustration principale](#) par [Luca Bravo](#) sur [Unsplash](#)