

## Intitulé : Programmer orienté objet en Java

### Programmer orienté objet en Java

#### Description du métier, de l'activité ou de la situation professionnelle à partir duquel le dispositif de formation visant la certification est initié

La compétence de programmation orientée objet en Java occupe une place prépondérante dans le domaine de l'informatique en constante évolution, offrant un ensemble d'outils puissants et flexibles pour résoudre des problèmes complexes dans le développement d'applications, la conception de systèmes ou la gestion de bases de données.

Programmer orienté objet en Java permet de créer des logiciels modulaires et évolutifs. En structurant le code en classes et en objets, le développeur peut organiser le programme de manière à ce qu'il soit plus facile à comprendre, à maintenir et à mettre à jour. Les fonctionnalités peuvent être encapsulées dans des objets, facilitant ainsi la réutilisation du code et la gestion des dépendances.

Java est également largement utilisé dans le développement d'applications d'entreprise en raison de sa portabilité et de sa plateforme indépendante. La programmation orientée objet en Java permet de concevoir des applications robustes et compatibles avec différents systèmes d'exploitation. Cela garantit une flexibilité maximale pour déployer des solutions logicielles dans divers environnements.

La gestion de la complexité des projets informatiques est un autre aspect crucial de la compétence en programmation orientée objet en Java. En décomposant un système en objets interconnectés, les développeurs peuvent aborder chaque composant de manière isolée, simplifiant ainsi la résolution des problèmes et la détection des erreurs. Cela améliore la qualité du code et accélère le processus de développement.

La programmation orientée objet en Java est essentielle pour la création d'interfaces utilisateur conviviales. La modélisation d'objets permet de représenter de manière intuitive les entités du monde réel, rendant ainsi les applications plus accessibles et conviviales pour les utilisateurs finaux contribuant à l'adoption rapide des logiciels et à la satisfaction de ces utilisateurs.

Programmer orienté objet en Java est aujourd'hui plus crucial que jamais dans le paysage professionnel de l'informatique, en raison de la demande croissante de professionnels possédant cette expertise comme :

- Développeur Logiciel Java
- Data scientist
- Architecte Logiciel
- Ingénieur en Systèmes d'Information
- Développeur Android
- Ingénieur en Sécurité Informatique
- Consultant en Technologies de l'Information

	REFERENTIEL DE CERTIFICATION <i>défini les critères et les modalités d'évaluation des acquis</i>	
	MODALITES D'EVALUATION	CRITERES D'EVALUATION
C.1 Utiliser l'environnement de développement intégré de l'entreprise (IDE : par exemple, Eclipse, IntelliJ...) pour développer une application informatique, sa documentation technique et ses tests.	<p><i>Pour chaque modalité d'évaluation, des aménagements de contenus, de temps et de matériel sont prévus afin de garantir une égalité de droits et de chances aux candidats en situation de handicap.</i></p> <p><b>C1 est transverse dans tous les devoirs sur machine, projets et TP notés.</b></p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Qualité du produit : lisibilité, indentation.</li> <li>- Utilisation d'identificateurs significatifs par rapport au métier ou à la technique.</li> <li>- Opérationnalité du programme (réalise la fonction attendue) : il fonctionne.</li> </ul>
C.2 Écrire un composant logiciel (classe Java) pour encapsuler des données techniques (adresse réseau, fichiers...) et métiers (par exemples : factures, bulletins de salaire, horaires de trains...) et les traitements relatifs à ces données (calcul d'un total de facture, calcul de trajet en train...) en assurant la cohérence interne à l'aide de tests unitaires et d'intégration tout au long de l'exécution de l'application pour assurer sa robustesse et faciliter sa maintenance.	<p><b>Les modalités d'évaluation de la compétence C2 :</b></p> <ul style="list-style-type: none"> <li>- <b>Écriture de code en temps limité en centre de formation</b> : écrire une classe représentant une notion métier ainsi qu'un petit programme instanciant cette classe (création d'objets) et utilisant ses fonctionnalités.</li> </ul> <p>Par exemple : représenter des cartes de cantines ayant un numéro, un nom de titulaire, un solde. Des méthodes doivent permettre de créditer et de débiter une carte lors d'un passage en caisse.</p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Le programme se compile sans erreur.</li> <li>- L'indentation du code respecte la logique du programme).</li> <li>- Les identificateurs sont significatifs (les noms donnés aux fonctions et variables doivent faciliter la lecture du programme).</li> <li>- Le programme créé est conforme à la spécification (énoncé du TP).</li> <li>- La classe par construction interdit la création d'objets dans un état incohérent et empêche l'évolution d'objets vers un tel état (par exemple, créer une carte de cantine sans nom de titulaire).</li> </ul>

**Intitulé : Programmer orienté objet en Java**

- **Écriture et dépôt du code d'un projet applicatif individuel** avec un sujet imposé.

Par exemple : le développement d'un jeu de patience Klondike (jeu de cartes).

Pour toutes les étapes, une nouvelle version peut être demandée pour tenir compte des remarques du correcteur.

-Première étape : écrire la classe représentant les cartes à jouer et la classe représentant un jeu de 54 cartes avec la possibilité de mélanger ce jeu.

-Deuxième étape : mise en œuvre de l'héritage et du polymorphisme pour modéliser les différents ensembles de cartes présents sur l'espace de jeu : la pioche, les colonnes du tableau, les pieux de fondation.

-Troisième étape : intégrer au projet une interface graphique fournie sous forme d'un package.

-Quatrième étape : réaliser une application complète utilisant les composants développés aux étapes intermédiaires. Écrire la boucle principale du projet, la documentation et la possibilité de sauvegarder une partie.

-Une soutenance en face-à-face avec l'enseignant : le code est regardé et commenté par l'enseignant et les choix de l'élève sont interrogés. Une nouvelle version peut être demandée pour corriger les défauts détectés.

## Intitulé : Programmer orienté objet en Java

<p>C.3 Développer une application informatique par raffinements successifs en appliquant les concepts clés de la programmation par objet (héritage, polymorphisme, interface) pour minimiser l'effort et le temps de développement et favoriser la réutilisation de composants logiciels.</p>	<p><b>La modalité d'évaluation de la compétence C3 :</b> Mise en œuvre de l'héritage et du polymorphisme pour modéliser les différents ensembles de cartes présents sur l'espace de jeu : la pioche, les colonnes du tableau, les pieux de fondation (2<sup>ème</sup> étape du projet).</p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Le programme se compile sans erreur</li> <li>- L'indentation du code respecte la logique du programme)</li> <li>- Les identificateurs sont significatifs (les noms donnés aux fonctions et variables doivent faciliter la lecture du programme)</li> </ul>
<p>C.4 Rédiger la documentation technique des classes composants l'application informatique au format Javadoc pour permettre la réutilisation et la maintenance du code produit et rendre les logiciels évolutifs (développement de nouvelles fonctionnalités et/ou correction de bugs)</p>	<p><b>La modalité d'évaluation de la compétence C4 :</b> Réalisation d'une application complète utilisant les composants développés aux étapes intermédiaires. Ecriture de la boucle principale du projet, la documentation et la possibilité de sauvegarder une partie (4<sup>ème</sup> étape du projet).</p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- L'indentation du code respecte la logique du programme).</li> <li>- Les classes contiennent une documentation au format Javadoc.</li> </ul>
<p>C.5 Concevoir, écrire et exécuter des tests unitaires en utilisant le framework de tests Junit afin d'assurer la conformité du code à ses spécifications et d'éviter les régressions (réapparition d'erreurs)</p>	<p><b>Les modalités d'évaluation de la compétence C5 :</b></p> <ul style="list-style-type: none"> <li>- <b>Devoir réalisé hors centre de formation :</b> écrire une fonction et les tests Junit de cette fonction. Une partie de ces tests est fournie avec l'énoncé, une autre partie doit être écrite par l'élève.</li> </ul> <p>Par exemple, la fonction testée est la recherche du second plus grand élément d'une liste.</p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- L'exécution des tests avec les résultats attendus.</li> <li>- La couverture des tests (tous les cas décrits dans les spécifications). Par exemple, les tests devront prévoir des cas tels que la liste vide ou la présence d'ex-aequo.</li> </ul>

## Intitulé : Programmer orienté objet en Java

	<p>Réalisation d'une application complète utilisant les composants développés aux étapes intermédiaires. Écrire la boucle principale du projet, la documentation et la possibilité de sauvegarder une partie (4<sup>ème</sup> étape du projet).</p>	
<p>C.6 Mettre en œuvre une bibliothèque Java existante identifiée en interne ou en externe (opensource ou commerciale) à partir de sa documentation technique afin d'utiliser ses fonctionnalités pour écrire de nouveaux programmes (produire des rapports en PDF à partir de données, charger et afficher des images...).</p>	<p><b>La modalité d'évaluation de la compétence C6 :</b>          Ecriture de la classe représentant les cartes à jouer et la classe représentant un jeu de 54 cartes avec la possibilité de mélanger ce jeu (1<sup>ère</sup> étape du projet).</p> <p><b>La modalité ci-dessous évalue les compétences C6, C7 et C9.</b></p> <ul style="list-style-type: none"> <li>- <b>Devoir réalisé hors centre de formation :</b> compléter un ensemble complexe de classes fournies en ajoutant de nouvelles classes et en enrichissant les classes fournies.          Par exemple, un système de réservation de chambres d'hôtel avec des classes représentant les hôtels, les chambres, les voyageurs, les réservations et les tarifs en fonction des dates. Ces classes utilisent des collections (map, liste, etc).</li> </ul>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Un jeu de tests sera automatiquement appliqué. Par exemple, réserver une chambre : vérification de la disponibilité et de l'exactitude du prix.</li> <li>- L'efficacité du code mis en œuvre (rapidité d'exécution, concision des programmes, économie des ressources utilisées).</li> </ul>

## Intitulé : Programmer orienté objet en Java

<p>C.7 Modéliser des objets complexes et structurés (facture référençant des produits, liste des personnels, descriptif de réseau ferroviaire...) en utilisant les collections de la bibliothèque standard Java afin de créer plus efficacement des représentations qui correspondent aux besoins métiers des entreprises.</p>	<p><b>Les modalités d'évaluation de la compétence C7 :</b></p> <ul style="list-style-type: none"> <li>- Ecriture de la classe représentant les cartes à jouer et la classe représentant un jeu de 54 cartes avec la possibilité de mélanger ce jeu (1<sup>ère</sup> étape du projet).</li> <li>- <b>Devoir réalisé hors centre de formation</b> : compléter un ensemble complexe de classes fournies en ajoutant de nouvelles classes et en enrichissant les classes fournies. Par exemple, un système de réservation de chambres d'hôtel avec des classes représentant les hôtels, les chambres, les voyageurs, les réservations et les tarifs en fonction des dates. Ces classes utilisent des collections (map, liste, etc).</li> </ul>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Le programme se compile sans erreur.</li> <li>- L'indentation du code respecte la logique du programme.</li> <li>- Les identificateurs sont significatifs (les noms donnés aux fonctions et variables doivent faciliter la lecture du programme)</li> <li>- Le programme créé est conforme à la spécification du projet de création d'applicatif.</li> <li>- Les classes par construction interdisent la création d'objets dans un état incohérent et empêche l'évolution d'objets vers un tel état (par exemple, une carte qui serait un 11 de trèfles ou d'un jeu avec 8 as).</li> <li>- Les classes contiennent une documentation au format Javadoc.</li> <li>- Le projet contient plusieurs packages séparant les classes métiers des classes d'interfaces.</li> <li>- L'interface graphique est fonctionnelle.</li> <li>- L'héritage et le polymorphisme sont utilisés.</li> <li>- L'utilisation des collections.</li> </ul>
<p>C.8 Écrire des programmes Java qui lisent ou écrivent dans des fichiers textes ou binaires les données manipulées (listes d'étudiants et leurs notes, relevés de températures associées à une station météo...) pour les sauvegarder (persistance) et les recharger ultérieurement</p>	<p><b>La modalité d'évaluation de la compétence C8 :</b></p> <p>Réalisation d'une application complète utilisant les composants développés aux étapes intermédiaires. Écrire la boucle principale du</p>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Prise en compte de l'accessibilité au programme pour les utilisateurs en situations de handicap.</li> <li>- Le programme se compile sans erreur.</li> </ul>

## Intitulé : Programmer orienté objet en Java

	<p>projet, la documentation et la possibilité de sauvegarder une partie (4<sup>ème</sup> étape du projet).</p>	<ul style="list-style-type: none"> <li>- L'indentation du code respecte la logique du programme).</li> <li>- Les identificateurs sont significatifs (les noms donnés aux fonctions et variables doivent faciliter la lecture du programme).</li> <li>- Les classes contiennent une documentation au format Javadoc.</li> </ul>
<p>C.9 Écrire du code Java s'intégrant dans une architecture logicielle préexistante en utilisant des patterns simples (Modèle Vue Contrôleur MVC, Commande...) en vue de participer à l'écriture de programmes au sein d'une équipe de développement.</p>	<p><b>Les modalités d'évaluation de la compétence C9 :</b></p> <ul style="list-style-type: none"> <li>- Intégration au projet d'une interface graphique fournie sous forme d'un package (3<sup>ème</sup> étape du projet).</li> <li>- <b>Devoir réalisé hors centre de formation</b> : compléter un ensemble complexe de classes fournies en ajoutant de nouvelles classes et en enrichissant les classes fournies. Par exemple, un système de réservation de chambres d'hôtel avec des classes représentant les hôtels, les chambres, les voyageurs, les réservations et les tarifs en fonction des dates. Ces classes utilisent des collections (map, liste, etc).</li> </ul>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Le programme se compile sans erreur</li> <li>- L'indentation du code respecte la logique du programme)</li> <li>- Les identificateurs sont significatifs (les noms donnés aux fonctions et variables doivent faciliter la lecture du programme).</li> </ul>

## Intitulé : Programmer orienté objet en Java

<p>C.10 Créer l'interface graphique (Swing ou JavaFX) d'une application Java, composée de fenêtres, de boutons, de champs de saisie et capable de réagir à des actions de l'utilisateur (souris, clavier) pour permettre les interactions de l'utilisateur et de l'application.</p>	<p><b>Les modalités d'évaluation de la compétence C10 :</b></p> <ul style="list-style-type: none"> <li>- Intégration au projet d'une interface graphique fournie sous forme d'un package (3<sup>ème</sup> étape du projet).</li> <li>- <b>Devoir réalisé hors centre de formation :</b> A partir d'un squelette d'application (codes à trous) mis à disposition des candidats, réalisation d'une interface pour un dictionnaire, les mots du dictionnaire étant affichés dans une liste avec un champ de saisie pour les filtrer, l'utilisateur pouvant sélectionner un mot et visualiser sa définition.</li> </ul>	<p><b>Les critères d'évaluation sont :</b></p> <ul style="list-style-type: none"> <li>- Le programme se compile sans erreur.</li> <li>- L'indentation du code respecte la logique du programme).</li> <li>- Le programme créé est conforme à la spécification du projet de création d'applicatif.</li> <li>- Le programme réagit de façon cohérente (par exemple : un mot sélectionné fait apparaître sa définition et pas une autre), prédictible (conforme aux attentes de l'utilisateur).</li> </ul>
---	---	---