

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 1 - Hiver2021](#)

Commencé le mardi 16 février 2021, 14:14

État Terminé

Terminé le mardi 16 février 2021, 14:44

Temps mis 30 min 7 s

Note 13,00 sur 20,00 (65%)

Question **1**

Terminer

Note de 5,00 sur 5,00

Nommez une différence entre les concepts d'architecture et de conception. Pour chaque élément donné, indiquez s'il est pertinent pour l'architecture ou la conception.

Classe

Niveau (Tier)

Base de données

Héritage

Sécurité

Classe : Conception

Niveau: Architecture

Base de données: Architecture

Héritage: Conception

Architecture: Architecture

Différence: l'architecture réponds au "quoi?". Donc, ça réponds à quels paquetages, composantes et relations que nous voudrons avoir pour le système tandis que la conception consiste au "comment" nous allons concrètement implémenter l'architecture.

Commentaire :

Question **2**

Terminer

Note de 5,00 sur 5,00

Dans vos propres mots, définissez le clonage ou la duplication de code. Quel problème cela pose-t-il lors de la maintenance?

La duplication de code consiste à prendre une certaine logique qui est réutilisable et que, au lieu de l'abstraire par exemple dans une méthode pour la rappeler, on copie colle les lignes partout où c'est utilisable. Cela cause un problème pour la maintenance, car s'il y a une erreur dans la logique, il faudra retrouver tous les endroits où le code a été copié pour corriger l'erreur au lieu de le corriger à un seul endroit et cela peut introduire des erreurs humaines (un développeur a oublié de corriger l'erreur à un endroit). Plus le système est complexe, plus il sera difficile de se rappeler exactement tous les endroits où le code dupliqué a été copié collé.

Commentaire :

Question 3

Terminer

Note de 3,00 sur 10,00

Vous recevez des métriques pour la classe `IncomeTaxCalculator` et ses méthodes. Sur la base des métriques, identifiez et nommez un antipatron présent. Justifiez votre réponse. Comment refactoriser le code pour supprimer l'antipatron?

| Class | Method | Weighted Method Count | Cyclomatic Complexity | Lines of CodeNesting | |
|----------------------------------|--------------------------------------|-----------------------|-----------------------|----------------------|---|
| <code>IncomeTaxCalculator</code> | | 26 | | 1345 | |
| | <code>readBasicIncome</code> | | 4 | 211 | 3 |
| | <code>calculateIncomeTax</code> | | 12 | 704 | 8 |
| | <code>readSupplementaryIncome</code> | | 7 | 318 | 3 |
| | <code>readBenefits</code> | | 3 | 112 | 1 |

| Class | Method | Weighted Method Count | Cyclomatic Complexity | Lines of CodeNesting | |
|----------------------------------|--------------------------------------|-----------------------|-----------------------|----------------------|---|
| <code>IncomeTaxCalculator</code> | | 26 | | 1345 | |
| | <code>readBasicIncome</code> | | 4 | 211 | 3 |
| | <code>calculateIncomeTax</code> | | 12 | 704 | 8 |
| | <code>readSupplementaryIncome</code> | | 7 | 318 | 3 |
| | <code>readBenefits</code> | | 3 | 112 | 1 |

Antipatron: Manque d'abstraction

Justification: Je pense qu'il y a un manque d'abstraction, car la classe `IncomeTaxCalculator` s'occupe de lire dans la base de données (je suppose vu le nom des méthodes) et de calculer l'impôt.

On devrait extraire la responsabilité de lire les valeurs ailleurs et que la responsabilité du `IncomeTaxCalculator` soit de calculer l'impôt. Je pense que le principe de SRP est violé et on peut aussi le voir avec les métriques. Lorsque les méthodes ont autant de lignes de code, il y a sûrement l'option de séparer et d'abstraire plus. De plus, la complexité cyclomatique est énorme et il y a beaucoup d'imbrications ce qui rends le code moins maintenable au final. Extraire des parties dans une autre classe sera que bénéfique pour réduire ces métriques et

Commentaire :

L'antipatron manque d'abstraction est pertinent aux données de la classe. Lorsque la fonctionnalité porte sur plusieurs responsabilités on parle de l'antipatron Multifaceted Abstraction.

Aussi, les métriques montrent d'autres antipatrons par rapport au code.

◀ [Déclaration d'honneur - Quiz 1 - Hiver2021](#)

Aller à...

[Declaration d'honneur - Quiz 2 - Hiver2021](#) ►

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 1 - Hiver2021](#)

Commencé le mardi 16 février 2021, 14:14

État Terminé

Terminé le mardi 16 février 2021, 14:44

Temps mis 30 min 2 s

Note 20,00 sur 20,00 (100%)

Question **1**

Terminer

Note de 5,00 sur 5,00

Nommez une différence entre les architectures en couches et multi-niveaux. Quel style est distribué et lequel est modulaire?

L'architecture multi-niveau est distribué.

L'architecture en couches est modulaire.

La différence entre les deux architectures est qu'avec les architectures en couche, on ne peut pas communiquer entre les couches qui sont au même niveau. Cependant, avec l'architecture multi-niveau, il est possible de communiquer entre chaque couche qui sont au même niveau.

Commentaire :

Question **2**

Terminer

Note de 5,00 sur 5,00

Dans vos propres mots, définissez le clonage ou la duplication de code. Quel problème cela pose-t-il lors de la maintenance?

La duplication de code est en gros du code qui est copié-collé à plusieurs places. En effet, cela peut avoir des répercussions lors d'une maintenance. Effectivement, les développeurs devront corriger la même erreur à toutes les places du code dupliqué. Cela peut devenir une tâche fastidieuse et on pourrait oublier à certaines places de corriger un bug qui a été réglé auparavant.

Commentaire :

Question 3

Terminer

Note de 10,00 sur 10,00

Vous développez un système de données pour les cas de COVID-19 dans la ville de Montréal. Le système stockera des données patient sensibles, accessibles uniquement à un nombre limité d'utilisateurs. Le nombre de caisses stockées est limité par la population de la ville, nous savons donc de combien de ressources nous avons besoin. Quel style architectural utiliseriez-vous pour implémenter ce système et pourquoi?

Selon l'énoncé, il est mentionné que nous voulons stocker les données des patients sensibles. En effet, c'est un principe de confidentialité. Donc, on peut penser à la sécurité du logiciel. Ensuite, on mentionne que les informations sont disponibles à un certain nombre de personnes. Ainsi, on peut penser à l'authenticité lorsqu'une personne tente d'accéder aux informations. Si elle ne fait pas partie des membres qui ont le droit d'accès le système refusera l'accès à cette dernière. Cela signifie que la sécurité est encore plus importante. Ainsi, on peut penser à une architecture du style CBA : Architecture des composantes. Aussi, au niveau des erreurs, il en faut le moins possible : reliability.

Commentaire :

[◀ Declaration d'honneur - Quiz 1 - Hiver2021](#)[Declaration d'honneur - Quiz 2 - Hiver2021 ►](#)

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 2 - Hiver2021](#)

Commencé le mardi 9 mars 2021, 14:12

État Terminé

Terminé le mardi 9 mars 2021, 14:41

Temps mis 29 min 55 s

Note 20,00 sur 20,00 (100%)

Question **1**

Terminer

Note de 5,00 sur 5,00

Dans vos propres mots, décrivez le fonctionnement de MapReduce. Quel style architectural suit-il?

MapReduce constitue un modèle de programmation pour l'analyse distribuée des mégadonnées. Elle se sépare en 2 parties: une tâche map et une tâche reduce. La tâche map constitue un travail d'analyse sur les données comme un tri ou une addition tandis que reduce représente un travail de sommation. Donc, sa tâche sera d'agréger les résultats des tâches map individuelles.

MapReduce suit l'architecture master-slave. Donc, le master s'occupe de distribuer les tâches et de balancer la charge entre les ouvriers. Les ouvriers peuvent accepter des tâches map ou des tâches reduce.

Commentaire :

Question 2

Terminer

Note de 5,00 sur 5,00

Dans vos propres mots, définissez les bases de données de documents. Nommez deux de leurs avantages.

Les bases de données "Document" traitent les données comme des fichiers qui peuvent avoir n'importe quelle taille et complexité, mais qui suivent un encodage spécifique (JSON, XML, etc). On peut demander des documents grâce aux clés.

Deux de leurs avantages sont:

- Si notre application ont des données qui sont des documents et que la structure peut varier dans le temps, cette base de données s'adapte très facilement.
- Avec cette base de données, on peut demander des documents en fonction de leurs attributs

Commentaire :

Question 3

Terminer

Note de 10,00 sur 10,00

Vous développez un système de stockage des dossiers des patients. La plupart des cliniques enregistrent les dossiers patient sous forme de feuilles de calcul. Chaque clinique peut avoir des colonnes spécifiques différentes, mais toutes ont les mêmes données générales. Quel type de base de données NoSQL utiliseriez-vous pour implémenter le système? Justifiez votre réponse.

Pour implémenter ce système, j'utiliserais une base de données Wide-column, car, puisque les clients enregistrent les dossiers sous formes de feuilles de calcul (je suppose comme Excel), Wide-column est le système qui se rapproche le plus de ces fonctionnements. Donc, pour le client qui communique avec l'interface, il ne sera pas complètement perdu puisqu'il assume l'existence d'un schéma et que Wide-column conserve ces concepts. En effet, il conservent les concepts de tableaux et de colonnes qui se retrouvent aussi dans les feuilles de calcul. De plus, la définition des colonnes sont très flexibles, ce qui est un point important pour ce système puisque chacune des cliniques peuvent avoir certaines colonnes spécifiques à elles. Ce système sera capable de supporter cela.

Commentaire :

◀ [Declaration d'honneur - Quiz 2 - Hiver2021](#)

Aller à...

[Declaration d'honneur - Quiz 3 - Hiver2021](#) ▶

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 2 - Hiver2021](#)

Commencé le mardi 9 mars 2021, 14:12

État Terminé

Terminé le mardi 9 mars 2021, 14:42

Temps mis 29 min 58 s

Note 19,00 sur 20,00 (95%)

Question **1**

Terminer

Note de 5,00 sur 5,00

Nommez la fonctionnalité qui rend Spark plus rapide que Hadoop. Expliquez pourquoi.

La fonctionnalité qui rend Spark plus rapide que Hadoop sont les RDDs. Ici, on parle du Resilient Distributed Dataset. En effet, Spark fonctionne en mémoire. En d'autres termes, Spark est efficace lorsqu'on parle de traitement rapide des données ainsi qu'un processus itératif. D'ailleurs, en mémoire, Spark réalise des tâches en mémoire 100 fois plus rapidement que Hadoop et 10 fois plus rapidement lorsque c'est sur disque. Effectivement, la cause est qu'on utilise moins d'écritures et de lectures par cycle.

Commentaire :

Question 2

Terminer

Note de 5,00 sur 5,00

Dans vos propres mots, définissez des bases de données de wide-column. Nommez deux de leurs avantages.

Les bases de données wide-column sont des bases de données de type NoSQL. Au niveau conceptuel, les wide-columns se rapprochent des bases de données relationnelles. Cependant, la notion de colonnes est assez flexibles. Effectivement, il est possible de regrouper des attributs dans des familles de colonnes. En quelques sortes, on pourrait considérer que les wide-columns sont des bases de données de type key-value en 2 dimensions. Il faut savoir qu'ils possèdent des colonnes et des rangées.

Au niveau des avantages, on pourrait penser au fait qu'ils peuvent supporter de grandes quantités de données. Aussi, la flexibilité des familles colonnes améliore la maintenabilité des données et la gestion des données variantes.

Commentaire :

Question 3

Terminer

Note de 9,00 sur 10,00

Vous développez un système de prévision météorologique. Le système reçoit les données météorologiques toutes les 30 minutes, il reformate les données et les stocke dans une base de données. Après une journée, le système utilise les données du jour pour mettre à jour un modèle de prédiction entraîné et stocke la prédiction mise à jour dans une base de données. Quel type d'architecture de traitement (Lambda ou Kappa) utiliseriez-vous et pourquoi? Décrivez l'architecture du système (vous pouvez fournir un diagramme).

L'architecture de traitement que je choisirais serait Lambda. En effet, on mentionne que les données sont reçues à chaque 30 minutes et elles sont traitées. Alors, l'architecture Lambda permet de traiter ces informations de façon efficace. Il faut savoir que ce n'est pas du vrai temps réel même si on reçoit des données à chaque 30 minutes. On doit avoir une architecture qui supporte bien le temps réel, mais pas que. Il faut des réponses rapides et il faut un système qui permet de supporter plusieurs mises à jour de façon efficace. Voici une architecture :

Commentaire :

On manque l'architecture mais la réponse est bonne.

[◀ Declaration d'honneur - Quiz 2 - Hiver2021](#)

Aller à...

[Declaration d'honneur - Quiz 3 - Hiver2021 ▶](#)

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 3 - Hiver2021](#)

Commencé le mardi 30 mars 2021, 14:13

État Terminé

Terminé le mardi 30 mars 2021, 14:43

Temps mis 30 min

Note 15,00 sur 20,00 (75%)

Question **1**

Terminer

Note de 3,00 sur 5,00

Nommez un défi de la modularité basée sur les bibliothèques. Comment les architectures orientées services résolvent-elles ce défi?

Un des problèmes de la modularité par des librairies est qu'il y a le partage de code. Ainsi, il y a des dépendances technologiques et temporelles.

Avec les bibliothèques on permet d'avoir des frontières de modularité plus douces. Aussi, les appels en mémoire sont plus rapides et moins chers.

Commentaire :

La question est "Comment les architectures orientées services résolvent-elles ce défi?". La deuxième partie de la réponse n'est pas correcte.

Question **2**

Terminer

Note de 5,00 sur 5,00

Décrivez dans vos propres mots l'antipatron de la loi sur le drapeau rouge (Red Flag Law). (La description relative au logiciel)

Pour représenter cet antipatron, on peut penser à des personnes qui devaient marcher devant des automobiles avec des drapeaux rouges. En effet, les automobiles devaient circuler à la même vitesse que les piétons. Au niveau des projets informatiques, cela signifie que les développeurs n'utilisent pas les bonnes pratiques des microservices. En effet, tout ce qui a attiré au processus en cascade, aux tests manuels et aux commits pendant des moments de maintenance fait en sorte que la vitesse d'efficacité est très lente. Effectivement, le but des microservices est de travailler sur différentes tâches afin d'être plus efficace et de mieux séparer les tâches. Pour qu'on puisse utiliser une architecture en microservice, il faut utiliser des pratiques DevOps pour apporter un côté d'automatisation (plutôt que de faire des tests manuellement), organiser les équipes adéquatement et utiliser le modèle Strangler Application.

Commentaire :

Question 3

Terminer

Note de 7,00 sur 10,00

Vous développez un portail de statistiques sportives. Les utilisateurs peuvent accéder aux statistiques du basketball, du hockey, du baseball et du football. Ils peuvent voir les statistiques des saisons actuelles ou passées, pour des équipes spécifiques ou comparer des joueurs. Comment implémenteriez-vous ce système? Quels services développeriez-vous? Comment organiseriez-vous vos données? Comment organisez-vous votre équipe de développement?

Pour ce système, on pourrait utiliser une architecture en microservices. En effet, nous aurions un service pour la gestion des statistiques des différents sports. Pour chaque sport on pourrait avoir une architecture en microservice. En effet, un microservice pour les statistiques de basketball, un autre pour le hockey, etc. On aurait un autre microservice pour la comparaison des joueurs. Finalement, le dernier microservice serait en lien avec les statistiques des différentes équipes pour chaque sport.

Au niveau de la gestion de l'équipe, on aurait une équipe front-end, une équipe back-end, des testeurs pour le back-end, des testeurs pour le front-end, une équipe d'assurance qualité et des intégrateurs systèmes externes. On limiterait le nombre de personnes dans chaque équipe à 12 maximum.

Au niveau des données, on utiliserait une base de données. On pourrait opter pour une base de données NoSQL. À cet effet, les documents seraient une bonne option car on aurait beaucoup de données à traiter.

Commentaire :

- Plusieurs services par sport? C'est un risque de nanoservices.
- Une seule BD pour tous les sports? Est-il efficace?

◀ [Declaration d'honneur - Quiz 3 - Hiver2021](#)

Aller à...

[Declaration d'honneur - ExamenFinal - Hiver2021](#) ▶

[Tableau de bord](#) / [Mes cours](#) / [LOG8430 - Architecture logicielle et conception avancée](#) / [Quiz and Exams](#) / [Quiz 3 - Hiver2021](#)

Commencé le mardi 30 mars 2021, 14:14

État Terminé

Terminé le mardi 30 mars 2021, 14:44

Temps mis 30 min 3 s

Note 19,50 sur 20,00 (98%)

Question **1**

Terminer

Note de 5,00 sur 5,00

Nommez un avantage et un désavantage des services SOAP.

un grand avantage d'utiliser SOAP est qu'il est indépendant du protocole de communication. Par contre un désavantage de ce type de service est qu'il n'est pas efficace

Commentaire :

Question **2**

Terminer

Note de 5,00 sur 5,00

Comment les microservices sont-ils conçus pour l'échec? Décrivez le concept dans vos propres mots.

Les microservices sont conçus pour l'échec en faisant en sorte d'être capable de réduire l'impact sur l'utilisateur si un microservice échoue. De plus, on peut effectuer la pratique de l'injection aléatoire, inattendue et intentionnelle de pannes dans les services, les nœuds ou même les centres de données pour voir comment les différents microservices réagissent. Cela permet de vérifier le comportement en cas d'échec et de s'assurer que l'impact de l'échec sera minimale sur l'utilisateur.

Commentaire :

Question 3

Terminer

Note de 9,50 sur 10,00

Vous développez un système de gestion de la base de données d'une librairie électronique. Le système sera utilisé par les administrateurs et les utilisateurs. Un administrateur peut uniquement ajouter, supprimer ou modifier des livres et leurs informations. Un utilisateur peut parcourir des catalogues, rechercher des livres par catégorie ou par auteur, commander ou réserver un livre. Concevoir et décrire une architecture SOA pour ce système. Vous pouvez combiner plusieurs styles architecturaux pour votre implémentation.

Pour ce système, nous pouvons utiliser une architecture REST qui aura plusieurs routes pour obtenir les informations souhaités. Avec REST, nous pouvons aussi effectuer de la pagination pour la recherche de livres. Donc, ce système aura un client avec un serveur REST ayant ces différentes routes

pour l'administrateur:

- `api/v1/book`

Pour cette route, les verbes GET et POST seront supportés. GET permet d'avoir l'ensemble des livres. Cette route supporte la pagination pour parcourir le catalogue de livres. POST permet de créer une nouvelle entrée dans la base de données.

- `api/v1/book/{id}`

Pour cette route, les verbes GET, PATCH et DELETE seront supportés. GET permet d'obtenir un livre précis dans la base de données. DELETE permet d'effacer un livre en particulier. PATCH permet de modifier/remplacer un livre précis dans la base de données.

Pour l'utilisateur:

- `api/v1/book`

Pour cette route, les verbes GET seront supportés. GET permet d'avoir l'ensemble des livres. Cette route supporte la pagination pour parcourir le catalogue de livres.

Pour cette route, il y aura des paramètres optionnels qui permettront de filtrer la liste des livres à retourner: *author* pour filtrer selon l'auteur et *category* pour filtrer par categorie.

- `api/v1/book/{id}`

Pour cette route, les verbes GET seront supportés. GET permet d'obtenir un livre précis dans la base de données.

- `api/v1/book/{id}/reserve`

Ici, que le verbe POST sera supporté. Celui-ci enverra toutes les informations nécessaire pour réserver le livre en question

- `api/v1/book/{id}/order`

Ici, que le verbe POST sera supporté. Celui-ci enverra toutes les informations nécessaire pour acheter le livre en question

Il faut aussi noter qu'il faudra ajouter un système d'authentification au sein du système et aussi d'implémenter cela dans l'architecture REST. C'est-à-dire, si un utilisateur effectue une requête, de vérifier que cette personne a les droits pour effectuer la requête avant de l'exécuter. Sinon, retourner le code d'erreur HTTP approprié.

Commentaire :

Bonne réponse. L'interface utilisateur pourrait être RPC/SOAP.

◀ [Declaration d'honneur - Quiz 3 - Hiver2021](#)

Aller à...

[Declaration d'honneur - ExamenFinal - Hiver2021](#) ▶