Commencé le	jeudi 17 novembre 2022, 23:07
État	Terminé
Terminé le	jeudi 17 novembre 2022, 23:25
Temps mis	18 min 11 s

Question **1**

Correct

Noté sur 0,33

Sélectionnez toutes les bonnes réponses:

- [ ] a. La journalisation rend le code source plus facile à lire
- [x] b. La journalisation peut être utilisée pour l'assistance d'utilisateur/client ✔
- [x] c. La journalisation a des impacts sur la performance du système quand elle est utilisée ✔
- [ ] d. La journalisation est un moyen d'analyser statiquement nos systèmes logiciels
- [ ] e. Il est facile de déterminer a priori quels logs nous aideront à rendre nos systèmes plus robustes

Votre réponse est correcte.

Question **2**

Correct

Noté sur 0,33

Indiquez lesquelles des affirmations suivantes sont vraies et lesquelles sont fausses:

Les logs sont par définition des messages structurés qui contiennent des horodatages (timestamps): Faux ✔

Le débogage (DEBUG) est le seul niveau de journalisation utile pour tester nos logiciel: Faux ✔

La journalisation peut autant être utilisée pour déboguer que simplement pour obtenir des informations sur l'utilisation du système: Vrai ✔

Dans les pratiques de journalisation modernes, l'exécution d'un système et l'enregistrement des logs se produisent sur la même machine: Faux ✔

Les logs au niveau du système sont normalement plus petits (en KB/MB/GB/TB) que les logs au niveau de l'application: Vrai ✔

Votre réponse est correcte.

Question **3**

Correct

Noté sur 0,33

Voici un exemple de logs structurés:

```
import datamanipulation
import datetime

def main():
    log_pour_debug("Starting the program")
    data = datamanipulation.load_data()
    log_pour_debug("data loaded")

    time = str(datetime.now(datetime.timezone("EST")).isoformat())
    log_pour_debug(f"Starting data aggregation at: {time}")
    new_data = datamanipulation.aggregate_data(data)
    log_pour_debug(f"Finished data aggregation at: {time}")

    log_pour_debug("Saving the data")
    datamanipulation.save_data(new_data)
    log_pour_debug("Exiting the program")

def log_pour_debug(log_message):
    print(f"DEBUG: {log_message}")

if __name__ == '__main__':
    main()
```

Veuillez choisir une réponse.

- Vrai
- Faux ✔

Un log structuré n'utiliserait pas de phrases complètes ni de messages écrits "à la main", mais uniquement des données d'application. Ce type de log nécessite de filtrer les messages avec des regex ou autre ce qui est très fastidieux. De plus, étant donné que toutes les logs n'ont pas d'horodatage (de timestamps), il est impossible de déterminer la performances de certaines étapes.

Question **4**

Correct

Noté sur 1,00

Veuillez télécharger les référentiels suivants (nous les utiliserons en classe) :

https://github.com/logpai/logparser

Le fichier: Linux_2k.log ici:https://github.com/logpai/loghub/tree/master/Linux

Le fichier: Android_2k.log ici: https://github.com/logpai/loghub/tree/master/Andriod

Vous pouvez simplement mettre la réponse à vrai pour obtenir le point.

Veuillez choisir une réponse.

- Vrai ✔
- Faux

◄ Vidéo C11 - Journalisation de Logiciel

Aller à…

Diapos Vidéo ►