

TD4 INFO0306 : Programmation mobile

Sujet : Persistance des données dans Android

1. QCM

- Détaillez vos réponses.
- Aucune ou plusieurs réponses possibles.

Question 1 : La persistance des données...

- A : s'effectue toujours avec les préférences partagées.
- B : peut être effectuée sur le stockage interne.
- C : peut se faire avec une base de données.
- D : peut être réalisé sur le web.

Question 2 : Les préférences utilisateurs...

- A : sont sauvegardées dans un fichier XML.
- B : sont sauvegardées automatiquement.
- C : sont sauvegardées manuellement.
- D : sont accessibles avec `getPreferences(Context.MODE_PRIVATE)`.

Question 3 : Pour les préférences,

- A : on préférera un fichier XML schématisant le contenu de la vue préférences.
- B : on peut utiliser un fichier Java et la classe `SharedPreferences`.
- C : permet de sauvegarder tous les types de données.
- D : fonctionne avec un mécanisme clé/valeur.

Question 4 : Un fichier

- A : peut être placé dans le cache externe.
- B : d'une application est toujours effacé si l'application est supprimée.
- C : peut être lu sans permission sur le stockage externe.
- D : peut être sauvegardé uniquement si le stockage externe est disponible.

Question 5 : Une base de données prend en charge...

- A : le type de données `INTEGER` (entiers).
- B : le type de données `REAL` (réels).
- C : le type de données `TEXT` (textes).
- D : le type de données `NULL`.

Question 6 : Sous Android,

- A : la base de données est commune à toutes les applications.
- B : une base de données est prévue et est unique pour chaque application.
- C : on exploite une base de données directement avec `SQLiteDatabase`.
- D : on utilise une classe d'aide facilitant l'usage d'une base de données `SQLiteOpenHelper`.

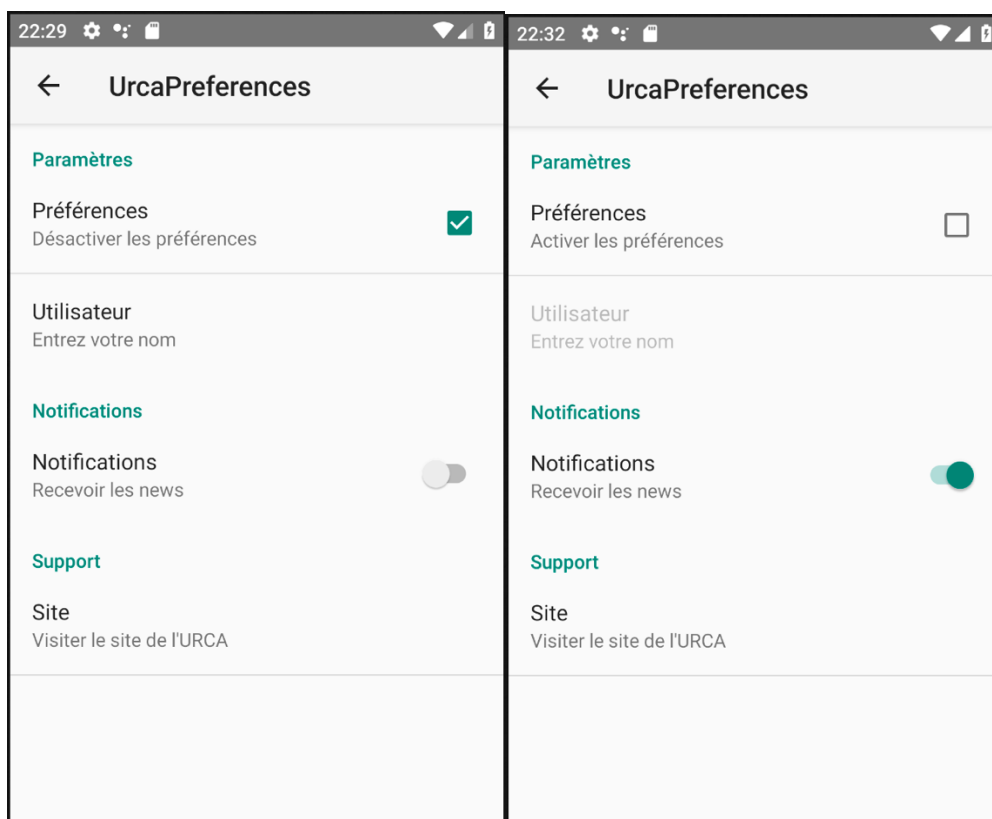
Question 7 : DAO...

- A : est un design pattern.
- B : permet de faire le lien entre la couche matérielle et les données.
- C : est obligatoire pour ajouter, supprimer, lire ou modifier des données dans une base.
- D : accède à une base de données avec la méthode `open()`.

2. Préférences partagées

Question 8 : Expliquez à quoi sert l'attribut `android:key`. Que se passe-t-il on oublie de le mettre ?

Question 9 : On souhaite obtenir les préférences utilisateur suivantes. Comment procéderez-vous ?
Écrire le code XML correspondant ?



3. Fichiers

Question 10 : Écrivez une méthode permettant d'écrire dans un fichier « urca.txt » le mot « Urca ». Ce fichier sera sauvegardé sur le stockage interne et se supprimera à la suppression de l'application.

Question 11 : Écrivez une méthode capable de lire le fichier « urca.txt » et d'afficher son contenu dans un Toast.

Question 12 : Modifier la méthode d'écriture pour que chaque clic sur le bouton ajoute le mot « Urca » au fichier. Pour obtenir une séquence de mot « Urca – Urca – Urca... »

Question 13 : Écrivez une méthode capable de supprimer le fichier « urca.txt » uniquement s'il existe.

Question 14 : Écrivez une méthode capable de vérifier si le stockage externe est disponible en lecture/écriture avec un message sous forme de Toast.

Question 15 : La même chose en lecture seule.

4. Base de données

On souhaite réaliser une application capable d'enregistrer des étudiants dans une table « student ». Chaque étudiant dispose d'un nom et d'un prénom.

Question 16 : Rappelez le principe de fonctionnement d'un objet *SQLiteOpenHelper* et écrivez une classe *UrcaDBHelper* dérivant de *SQLiteOpenHelper*.

Question 17 : Écrivez une méthode permettant de retourner un objet *SQLiteDatabase* en mode lecture seule puis une autre méthode pour le mode en lecture/écriture.

Question 18 : En cas de mise à jour de la base de données (dans les deux sens) on souhaite recréer la table « student ». Écrivez le code correspondant.

Question 19 : Écrivez une méthode permettant d'ajouter un étudiant dans la base. On supposera la présence de deux zones de texte (nom et prénom).

Question 20 : Écrivez une méthode capable de récupérer un étudiant aléatoirement et d'afficher son nom et prénom dans un Toast.

Question 21 : Écrivez une méthode capable de récupérer l'ensemble des étudiants et de les afficher un par un dans le LogCat.