# Annexe 3B– Sérialisation

Résumé de l'Annexe précédente…

On peut utiliser le Bundle savedInstanceState pour conserver les infos tant que l'app n'est pas complètement fermée ( recevoir un appel, tourner le téléphone, faire Home,etc…)

Cependant, la sérialisation peut être un autre moyen pour préserver des informations au-delà de la fermeture de l'App. Faites l'exemple suivant pour vous en convaincre:

## Exercice

On va développer une app permettant de lire des pensées inspirantes à portée de main. L'app fera apparaître une pensée au démarrage ou suite au clic du bouton "Une autre…".

L'usager peut aussi ajouter des pensées et elles deviendront des pensées à l'intérieur de la rotation. ( voir captures d'écran aux pages suivantes )

1. Utilisez le projet présent dans LÉA comprenant la classe ListePensees.. Cette classe n'est pas complète.

La classe comprend deux variables d'instance dont une variable de type Context. À quoi peut apparente-t-on un objet Context habituellement ? ( Encerclez la bonne réponse )

1. À un Singleton
2. À une activité
3. À un Intent
4. À un fichier de positionnement XML

Ajouter le code nécessaire pour transformer la classe ListePensees en Singleton de la façon vue en classe. Faites-en sorte d'initialiser la variable context par un paramètre et d'initialiser la variable Vector par un nouveau Vector auquel vous ajouterez comme première pensée : "*Vaut mieux avoir du cœur que d'avoir raison*" .

1. Codez les méthodes suivantes à l'intérieur de votre classe ListePensees destinée à être un Singleton :

* Une méthode permettant d'ajouter une Pensée à votre Vector
* Une méthode permettant de récupérer une Pensée au hasard parmi celles présentes dans le Vector
* Une méthode permettant de retourner le Vector ( get )

1. Attendez les infos du prof afin de compléter les deux méthodes partiellement codées consistant à écrire le Vector dans le fichier de sérialisation et à lire cet objet dans le fichier de sérialisation pour le récupérer suite à une fermeture de l'app ou le départ d'une activité. On précéde les méthodes openFileInput et openFileOutput du context étant donné le résultat de la question A.

Soyez conscients des exceptions que les appels de openFileInput / openFileOutput peuvent causer.

## Notes du prof

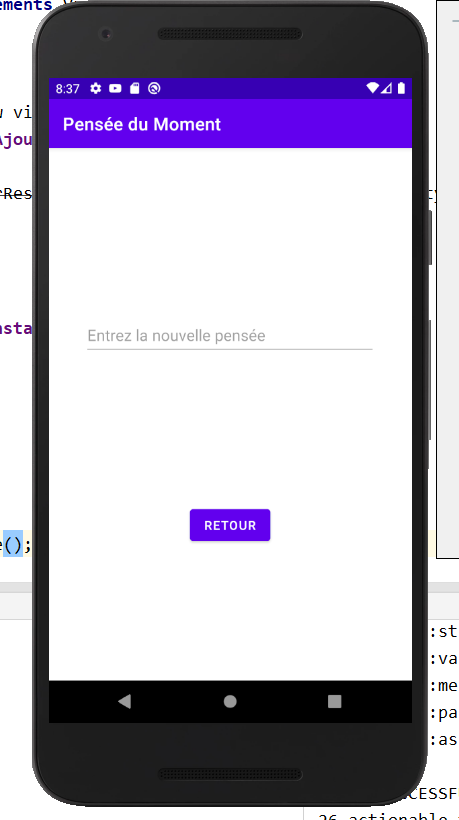
Qu’est-ce que la sérialisation ?

* Écrire dans un fichier de sérialisation
* Flux de traitement (buffer) spécial
* On peut sérialiser plusieurs objets dans le même fichier

D) Codez à présent votre Activité initiale, déjà dessinée :



* Faites en sorte qu'au démarrage, l'app choisisse une pensée au hasard ( au début il y en aura une seule )
* Un clic sur une autre… en choisira une autre à partir de la méthode que vous avez codée au # précédent.
* Un clic sur le bouton Ajouter permettra d'afficher **une autre activité où l'usager sera incité à écrire une nouvelle pensée pour ensuite revenir sur l'activité courante 🡪 procédé startActivityForResult de l'annexe 3**



**De retour sur l'activité courante,** vous pouvez utiliser un Toast pour signifier que la nouvelle pensée a bien été enregistrée et faisant maintenant partie de la liste des pensées.

Voici le code pour coder un Toast si vous l'avez oublié :

Context context = getApplicationContext();  
CharSequence text = **"texte présent"**;  
**int** duration = Toast.***LENGTH\_SHORT***;  
  
Toast toast = Toast.*makeText*(context, text, duration);  
toast.show();

Faites-en sorte qu'on puisse conserver la liste au-delà de la fermeture de l'application à l'aide des méthodes codées dans la question C. Si on ferme l'application avec le bouton triangle du téléphone Nexus, on peut réutiliser l'app sans problème et avec les mêmes pensées du moment.

**Des exemples de pensées :**

*Les sciences sont des serrures dont la clé est l‘étude. ( Inconnu )*

*La science consiste à passer d‘un étonnement à un autre ( Aristote )*

*Aimer c'est savoir dire je t'aime sans parler ( Victor Hugo )*

*On ne voit bien qu'avec le cœur, l'essentiel est invisible pour les yeux ( St-Exupéry )*