

ANDROID - TP1

Source et pdf de ce TP :
<https://bitbucket.org/olevitt/technologies-mobiles>

Ce TP a deux objectifs :

- Se familiariser avec le SDK android et le plugin ADT pour eclipse
- Construire une petite application de simulateur de crédit

1 Mise en place

1.1 Verifier l'installation

- Lancer Eclipse
- Vérifier dans **Window/Preferences** la présence du menu android
- Vérifier que le chemin du SDK est bien rempli

1. Quels sont les niveaux d'API disponibles ?

1.2 Hello world

- Créer un nouveau projet android via **File/New/Other/Android/Android Project**
- Executer l'application (Ctrl + F11)

2. Pourquoi l'application ne se lance t'elle pas ?

- L'android virtual device manager s'ouvre et permet de créer un émulateur
- L'AVD manager est aussi disponible via **Window / Android virtual device manager**

1.3 Créer un appareil virtuel

- Créer un nouvel appareil avec le plus grand niveau d'API possible
- Choisir x86 plutôt qu'ARM si possible
- Démarrer l'appareil
- Manipuler un peu l'émulateur

3. Quelles sont les applications pré-installées sur l'émulateur ?

4. Comment voir la log de l'émulateur ?

2 Construire notre simulateur de crédit

2.1 Un peu d'UI

Modifier le layout main pour y ajouter quelques composants :

- Un champ de texte (EditText) : montant
- Un champ de texte (EditText) : durée
- Ajouter un bouton (Button) : calculer
- Obliger l'utilisateur à ne rentrer que des chiffres dans les champs en utilisant la propriété InputType

5. Que faut t'il prévoir pour pouvoir utiliser ces views en java ?

2.2 Rendre l'interface vivante

- Ecouter les clicks sur le bouton et déclencher un toast lors d'un click sur le bouton
- Rappel : 2 techniques ont été vues en cours pour faire ça
- Dans le toast, afficher le contenu du champ de texte (editText.getText())

2.3 Implémenter la logique métier

- Lors d'un click sur le bouton, récupérer les valeurs de montant et durée
 - Aide : Integer.parseInt(String) permet de convertir une chaîne en entier
 - Implémenter la logique métier pour calculer les mensualités (taux fixe à 4% ou variable en fonction de la durée)
 - Afficher ce résultat d'abord dans un toast puis dans une view (TextView)
6. Comment cacher la view contenant les mensualités tant qu'elles n'ont pas été calculées ?

3 Aller plus loin

3.1 Partager le résultat

- Ajouter un bouton : partager
 - Mettre une icône de partage à ce bouton (`setBackgroundResource`)
 - Lors d'un click sur le bouton, lancer un intent implicite pour partager le résultat
7. Comment filtrer les activités proposées pour le partage ?
 8. Que se passe t'il si aucune activity n'est capable de répondre à l'intent ?

3.2 Créer un historique des calculs

9. Quelle solution de stockage de l'historique des calculs vous paraît la mieux adaptée ? (Rappel : on a le choix entre préférences, BDD SQLite et fichier plat)
 - Mettre en place la solution choisie
 - A chaque calcul, rajouter un élément à l'historique

3.3 Afficher l'historique

- L'historique sera affiché dans un écran séparé
 - Créer une nouvelle Activity qui contiendra l'historique (penser à la déclarer dans le Manifest)
 - Quel ViewGroup vous paraît le plus adapté pour afficher l'historique (nombre indéfini d'éléments)
10. Comment lancer cette activité ?