

# Soutenance de stage

## Résumé

Lundi 22 juin, c'est l'heure des soutenance de stage. Cette année est une année faste. Tous les étudiants ont rendus leur rapports dans les temps et les petites difficultés habituelles se sont vite réglées avec une forte participation des étudiants aux compte-rendus hebdomadaires. Bref, il ne reste plus qu'à admirer les étudiants en train de décrire comment ils ont convaincu les entreprises de leur savoir faire et de leur savoir être.

Du côté enseignant, il faut aussi respecter la forme, recueillir les appréciations des professionnels, noter les rapports, évaluer les soutenances et calculer la note selon le barème habituel.

Cette année nous avons une nouveauté. Chaque jury est doté d'une tablette qui remplace la calculatrice que l'on a jamais sous la main et permet de remonter au jury final les observations sur le stage, les entreprises et l'étudiant. Bon, cette année, je ne sais pas qui a fait ce logiciel (surement pas un étudiant de DUT), mais ce n'est pas au point : aucun calcul ne se fait et le chronomètre reste obstinément muet. Peux-tu nous aider ?

De fait :

- L'usage des documents internet est autorisé.
- L'usage des documents papiers est autorisé.
- L'usage d'un appareil android personnel est toléré.

## 1 Modalités du contrôle

Le résultat de votre travail devra être remis, sous forme d'un fichier compressé contenant l'intégralité de votre projet. Les étapes de résolution (correspondant aux questions) seront marquées par le gestionnaire de version GIT (la procédure est rappelée ci-dessous). Les étapes seront mentionnées dans le sujet sous cette forme :

Etape X : étiquette du commit.

### 1.1 Mise en place

Pour gagner du temps, nous vous fournissons le squelette de l'application avec en particulier les layouts et quelques ressources déjà construits. Votre travail consistera donc essentiellement à compléter le code java et/ou les ressources pour obtenir les fonctionnalités demandées. Il vous suffira d'importer dans votre répertoire **AndroidStudioProjet** le dossier que vous obtenez par recopie depuis `/home/public/place/M4104C`. La ligne de commande proposée est la suivante (si vous êtes placé dans votre répertoire **AndroidStudioProjects** :

```
cp -r /home/public/place/M4104C votrelogin
```

La description des éléments prédéfinis dont vous aurez besoin se trouve en annexe.

### 1.2 Procédure :

Le projet GIT est déjà initialisé au départ. Il vous suffira donc de garder un terminal dans le répertoire de votre projet et de lancer, à chaque étape, les commandes :

```
git add .
```

```
git commit -m 'étiquette'
```

### 1.3 Restitution

Compressez le dossier de votre projet et postez le sur moodle.

## 1.4 Seulement si vous préférez Eclipse...

Créez un projet par défaut avec une activité vierge, puis remplacez le contenu des fichiers `MainActivity.java`, `activity_main.xml`, `strings.xml`, `dimens.xml` et `colors.xml` par les contenus des fichiers correspondants publiés sur moodle.

Puis, dans un terminal, placez-vous dans le dossier de votre projet (ce devrait être dans votre workspace) et envoyez les commandes suivantes :

```
git init ; git add *.java ; git add *.xml ; git commit -m 'initial'
```

## 2 Calcul de la note finale.

La note finale est calculée par application des coefficients suivants (le total fait 100) :

- Note de rapport , coef. 40
- Note de soutenance (Oral), coef. 25
- Note d'entreprise, coef 35

Les notes attribuées peuvent être décimales (11,5 par exemple) et seront traitées comme des `double`.

La moyenne doit être affichée avec deux décimales (exemple : 10,00 ou 13,33 - peu importe que vous obteniez un point ou une virgule décimale).

### 2.1 test.

Les notes respectives de 10, 11 et 13 (rapport, oral, entreprise) devraient afficher une moyenne de 11,30.

Etape 1: Calcul.

## 3 Entrées illégales

Complétez le code pour que des entrées erronées (vides, négatives ou >20) donnent le message 'Erreur' comme note finale. Le libellé devrait être une ressource de nom `"erreur"`

Etape 2: Erreur.

## 4 Peuplement des listes déroulantes

Ajoutez les éléments (ressources et code) qui permettent aux 4 spinners d'afficher les valeurs **Très bien**, **Bien**, **Assez bien**, **Passable** et **Médiocre**, .

Etape 3: Adapter.

## 5 Chronomètre

Consultez la documentation sur le widget `chronometer`. Déduisez-en le code associée aux boutons **Start** et **Stop** pour obtenir les fonctionnalités suivantes :

- **Start**
  - Initialisation du chronométrage à zéro (cf. `elapsedRealTime`).
  - Démarrage du chronomètre.
- **Stop**
  - Arrêt du chronométrage à la valeur courante.

Etape 4: Chrono step 1.

## 6 Fonctionnement du chronomètre

Ajouter le code garantissant un usage alternatif des boutons **Start** et **Stop**. (**Start** ne peut être utilisé que quand le chonomètre est à l'arrêt, **Stop** n'est actif que quand le chronomètre est actif). Vous pouvez activez ou désactiver les boutons au moyen de la méthode `setEnabled`.

Etape 5: chrono step 2.

## 7 Tolérance aux changement de configuration

Si vous simulez un changement d'orientation (7 et 9 du pavé numérique), certaines valeurs sont remises à zéro et le chronomètre s'arrête.

Ajoutez les éléments (attributs, code, méthodes) qui permettent à l'application de conserver la continuité de com-  
portement (affichages, chronomètre, ...) en cas de changement de configuration.

Etape 6: Bundle.

## 8 Annexe : description du layout

