**Applis mobiles – Examen Final Formatif - 2H Nom :**

Redémarrez vos machines avant de commencer. Exercices et TP autorisés. Internet coupé sauf certains sites de documentation (Oracle, JSoup, MvnRepository, etc.).

Pour chaque élément de notation, si le code ne s’exécute pas ou ne s’arrête jamais, vous ne pouvez avoir plus que 40% du pointage. Si le résultat est codé en dur (*hardcodé*), vous ne gagnerez pas de points.

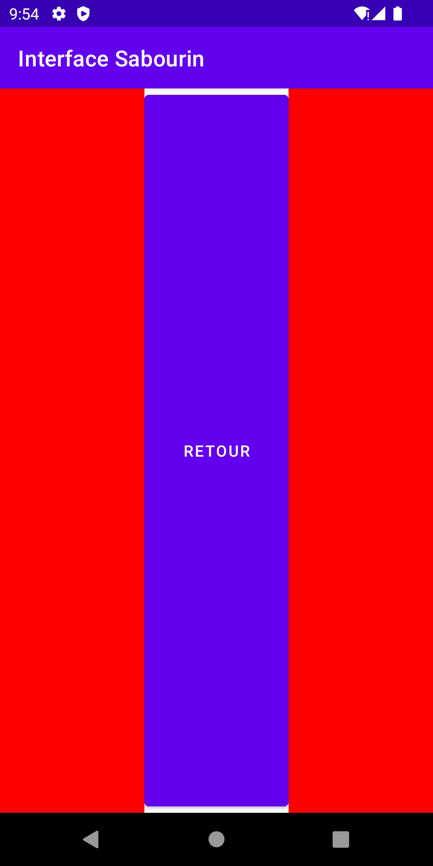
**Le final contiendra 3 exercices, ce formatif en contient 4. Faites un projet par exercice.**

**Interface graphique** 

En partant du projet de votre choix (pas de code fourni), vous devez programmer l’interface graphique suivante.

Activité 1

* **1 point :** Les titres des activités doivent correspondre, vous devez remplacer Deguet et Sabourin par votre nom de famille.
* **1 point :** Les étoiles sont centrées horizontalement.
* **1 point :** Le reste de la hauteur est divisé équitablement entre les 2 bandes noir-blanc blanc-noir. Les 2 bandes, le noir et le blanc ont la même largeur.
* **1 point :** Le bouton Final mène à la seconde activité. Le bouton Final ne prend pas toute la largeur, il est aligné à droite.

Activité 2

* **1 point :** Les bandes et le bouton ont tous la même largeur.
* **1 point :** Le bouton retour revient à la première activité.
* **1 point :** Le bouton occupe toute la hauteur.

**Code base de données**

En partant du projet de votre choix (pas de code fourni), vous devez fournir une application qui crée en base de données un objet.

* **1 point :** L’interface doit contenir un champ EditText et un bouton.
* **1 point :** Votre classe de modèle doit s’appeler FinalN*omFamille* en remplaçant nom de famille par ton nom de famille. Elle doit contenir 2 champs : nom de type String et id de type Long.
* **1 point :** L’appui sur le bouton doit créer un objet en BD qui contient le contenu du champ EditText.

## Tests

Vous trouverez un projet appelé Formatif\_Tests. Le projet contient une classe de service. Vous devez écrire les tests pour valider le bon fonctionnement de la méthode ajouterEvaluation. Chaque test doit indiquer clairement ce qui est testé par son nom de méthode.

VOUS NE DEVEZ PAS CORRIGER LE CODE, UNIQUEMENT ÉCRIRE LES TESTS POUR DÉTERMINER SI LE CODE EST BON OU PAS.

* **1 point :** Valider qu’on peut ajouter une évaluation avec succès.
* **1 point :** Valider que la méthode renvoie une EvaluationException lorsque le nom de l’évaluation n’est pas unique.
* **1 point :** Valider que la méthode renvoie une EvaluationException lorsque la note est inférieure à 1 ou supérieure à 255 ( 1 <= note <= 255).
* **1 point :** Valider que la méthode renvoie une EvaluationException lorsque le poids est inférieur à 1 ou supérieur à 100 ( 1 <= poids <= 100).

VOUS **NE DEVEZ PAS** MODIFIER LE **CODE**, UNIQUEMENT LES TESTS.

**RecyclerView**

À partir du projet de votre choix (pas de code fourni), vous devez fournir une application avec un RecyclerView selon les directives suivantes :

* **1 point :** Un bouton en haut avec « Ajouter ++ » comme texte.
* **1 point :** Le reste de l’activité est occupé par un RecyclerView.
* **2 points :** Appuyer sur le bouton déclenche l’augmentation d’un compteur (qui est initialisé à 0 quand votre appli démarre) puis ajoute la valeur du compteur dans la liste affichée (qui se met à jour).