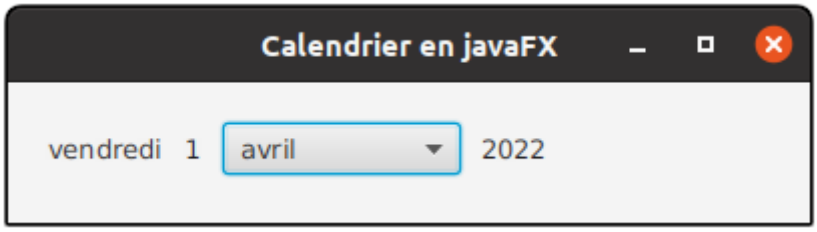
TD4 de développement d'application avec IHM

I) développement d'un calendrier

Le but de cet exercice est de développer un calendrier qui sera initialisé au départ à la date du jour. Nous utiliserons la classe **Calendar** du paquetage java.util (classe java)

1) développement de la vue



Il faudra au lancement de l'application, afficher la date du jour. Pour ceci, vous étudierez la classe abstraite **Calendar** => https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html

Vous avez aussi un tutoriel qui vous montre une utilisation de cette classe Java: https://devstory.net/10245/java-date-time.
Il faut regarder la partie 6 et la classe exemple : CalendarFieldsDemo.

Développer la méthode *update()* de la vue qui permet d'une part, d'obtenir via l'objet de type *Calendar* une date et d'autre part, de mettre à jour la vue.
L'objet de type *Calendar* est accessible car c'est un attribut de votre vue.

=> fichier **AppliCalendrier.kt**

2) modification de la date via un appui de touche

Lorsqu'on appuie sur les touches "left" ou "right" du clavier la date avance ou recule d'un jour. Il faut bien sûr que soit géré le changement de jour, de mois, d'année. Utilisez votre "instance" de **Calendar** et tout ceci sera géré automatiquement => **EcouteurToucheCalendrier.kt.**

On utilisera la méthode *set(…)* de *Calendar*. Par exemple :
calendar.set(Calendar.DAY_OF_MONTH, valeur) permet de modifier la valeur du jour de l'objet *calendar*.

3) modification du mois dans la comboBox

Quand on sélectionne un mois dans la comboBox alors le jour s'adapte. Ici, dans l'exemple de la capture d'écran, si on sélectionne *mai*, le jour devient *dimanche*. En utilisant l'objet de type **Calendar** a bon escient, il y a très peu de code à écrire.
=> **EcouteurComboBoxCalendrier.kt**

II) Développement d'un Chronomètre/Minuteur

1) Développement de la vue

Développer la vue suivante dans le fichier *AppliChronometreQ123.kt*

2) Vous disposez d'une classe **Timer** qui permet toutes les x secondes de générer un événement de type ActionEvent. En fait, c'est l'attribut de type **KeyFrame** qui génère cet événement. La classe **Timer** prend comme deuxième paramètre un gestionnaire d'événement qui sera déclenchée toutes les x secondes.

a) Instanciez un objet de type **Timer** dans votre vue pour qu'un événement soit généré toutes les secondes

b) Mise en place de l'écouteur

Vous implémenterez un seul écouteur auquel les composants seront abonnés => **EcouteurChronoQ123**

- lorsqu'on clique sur le bouton **Start**, le chronomètre démarre et le temps s'incrémente dans le TextField
- lorsqu'on clique sur le bouton **Stop**, le chronomètre s'arrête.
- Lorsqu'on clique sur le bouton **Reset** (il faut que le chronomètre soit arrêté), l'affichage redevient 0

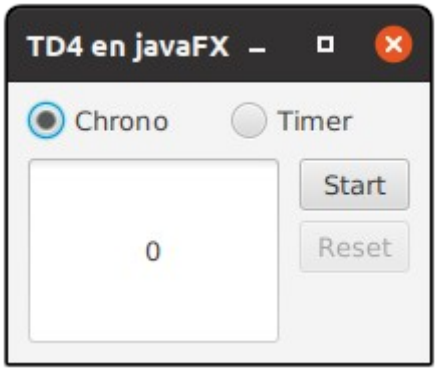
3) Il faut maintenant que certains boutons soient désactivés quand une action n'est pas possible :

- au départ, seul le bouton **Start** est activé
- lorsque le chronomètre est démarré, alors le bouton **Stop** est activé, le bouton **Reset** est désactivé.
- Lorsque le chronomètre est arrêté, le bouton **Start** et le bouton **Reset** sont activés

4) Mise en place d'un chronomètre/minuteur

a) modifiez la vue de l'application en ajoutant deux boutons radios et en supprimant le bouton **Stop**

=> code à développer dans AppliChronometreQ4.kt



b) développez l'écouteur auquel seront abonnés les composants dans **EcouteurChronoQ4**

- Lorsque le bouton radio nommé C*hrono* est sélectionné, l'application fonctionne en mode chronomètre.

- Lorsque le bouton radio nommé *Timer* est sélectionné, l'application fonctionne en mode minuteur (saisie d'une valeur au départ dans le TextField et décompte du temps jusqu'à 0).

- Le premier bouton change de nom en fonction des actions de l'utilisateur. Si le chronomètre ou le minuteur sont démarrés alors le bouton prend comme label "Stop" dans l'autre cas, il prend comme label "Start". Le comportement suite au clic sur le bouton est bien sûr différent en fonction de la valeur du label.

Gérez bien tous les cas possibles (on ne peut pas passer du mode Chrono au mode Timer sans que le chronomètre soit arrêté ...)