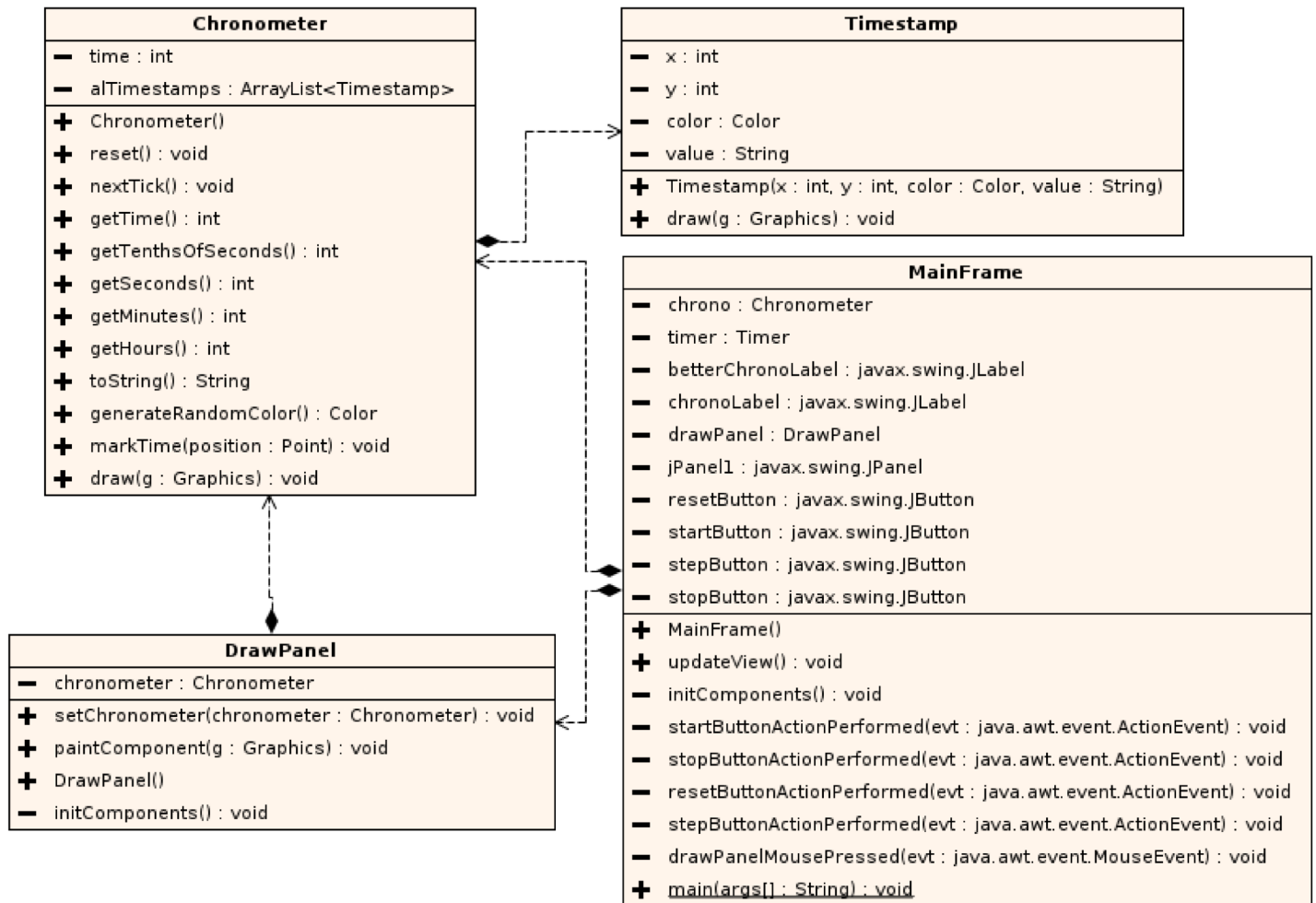
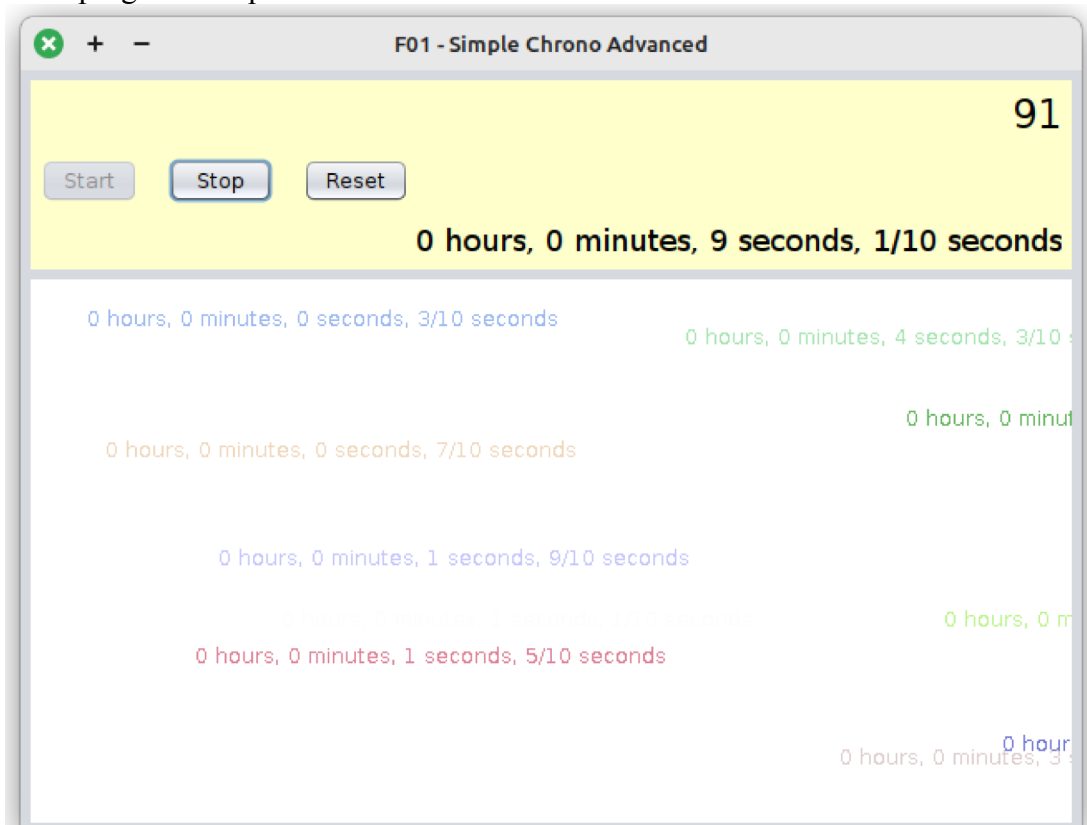


F01 – SimpleChrono – Version avancée (écrit en couleur le *timestamp* des clicks)

UML :



Dans la suite, vous allez développer un chronomètre (*Stoppuhr*) qui affiche le temps de manière textuelle. Quelques explications supplémentaires sur les classes et méthodes à programmer sont ci-dessous, ainsi qu'une capture d'écran du programme après lancement.



Attention – étapes de programmation :

1. commencez d'abord par l'affichage du temps sous forme textuelle
2. puis ajoutez la partie d'affichage des moments sur le canevas

→ ne programmez que les méthodes nécessaires au fur-et-à mesure !

Étape 1

La classe **Chronometer** implémente le chronomètre avec ses attributs et accesseur requis.

- l'attribut **time** représente le temps écoulé en dixièmes de secondes ;
- les méthodes **getHours**, **getMinutes**, **getSeconds** et **getTenthsOfSeconds** calculent, à partir de la valeur de **time**, la durée en heures, minutes, secondes et dixièmes de secondes.
Par exemple, pour représenter une durée de 3 minutes, 54 secondes et 5 dixièmes de secondes **time** vaut 2345. Dans ce cas **getMinutes** retourne 3, **getSeconds** retourne 54 et **getTenthsOfSeconds** retourne 5 ;
- la méthode **nextTick** fait avancer le temps d'un dixième de seconde ;
- la méthode **toString** retourne le temps écoulé sous forme textuelle comme montré ci-dessous ;
- la méthode **reset** remet **time** à 0.

La classe **MainFrame** contient l'attribut **time**, le chronomètre associé au bouton invisible **stepButton**, l'attribut **chrono**, qui représente le chronomètre en dixièmes de seconde et est directement instancié dès le départ. Les boutons de gestion (*start*, *stop* et *reset*) gèrent le démarrage, la pause et la remise-à-zéro du chronomètre. Note : ces boutons doivent être activés ou désactivés suivant s'ils sont utiles ou non. La méthode **updateView** met à jour l'affichage du temps.

Étape 2

La classe **Timestamp** représente un moment temporel avec sa position sur le canevas et la couleur de dessin.

La classe **Chronometer** implémente le chronomètre avec ses attributs et accesseur requis.

- l'attribut **allTimestamps** contient tous les moments à dessiner ;
- la méthode **generateRandomColor** crée et retourne une couleur (avec transparence) aléatoire ;
- la méthode **markTime** crée un nouveau moment (temps actuel) à la position passée en paramètre et l'ajoute à la liste ;
- la méthode **draw** dessine tous les moments contenus dans la liste **allTimestamps** ;
- la méthode **reset** doit aussi effacer tous les moments de la liste.

La classe **DrawPanel** est standard et dessine, si possible, tous les moments.

La classe **MainFrame** doit gérer un click sur le canevas en marquant le moment dans le chronomètre, ie. en ajoutant un nouveau moment. L'ajout ne peut se faire que lorsque le chronomètre tourne, pas lorsqu'il est arrêté ! La méthode **updateView** met aussi le canevas à jour.