

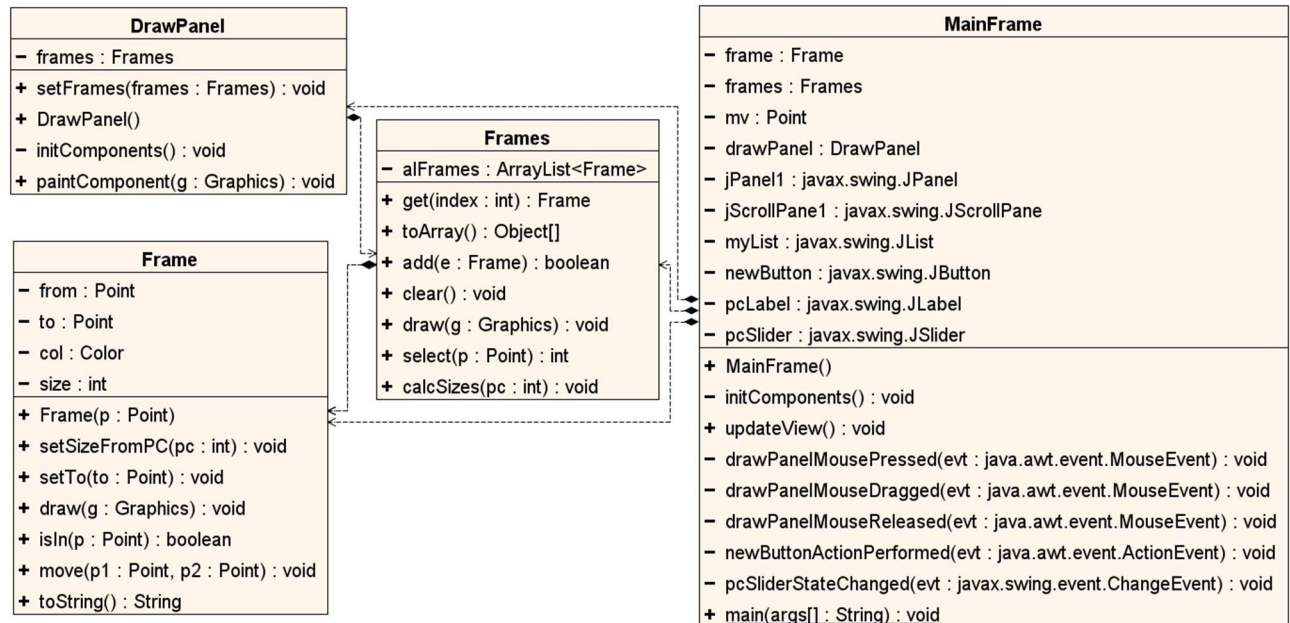
Respectez une **indentation** intelligente. Utilisez uniquement la fonctionnalité **JavaDoc @author**.

Le projet sert à gérer graphiquement des perles. Créez dans le lecteur Z: un dossier Dev1GIG_Frame_XXX où XXX est votre nom. Placez tous les fichiers dans ce dossier.

Pratique – NetBeans		60 pts
1.	Le projet DrawFrames contient le <i>JFrame</i> MainFrame , les classes Frame et Frames ainsi qu'un DrawPanel sur lequel on dessine et qui prend toujours la place maximale du MainFrame . Sur la fiche se trouve également à droite une <i>JList</i> , un slider, un bouton et un libellé.	
2.	La classe Frame (=cadre) possède les attributs from et to (deux points opposés de rectangle extérieur), col (initialisée avec couleur aléatoire), et size (un entier initialisé avec 0 qui désigne l'épaisseur des bords en pixels). Le constructeur permet de définir les deux points from et to à partir d'un seul paramètre. setTo() est un manipulateur de l'attribut to . La méthode setSizeFromPC(int pc) calcule la valeur de l'attribut size : On désigne d'abord le côté le plus petits entre largeur et hauteur du cadre. On calcule size en multipliant ce côté par pc et en divisant par 200. La méthode draw(...) dessine le cadre entouré de lignes noires. La méthode isIn(Point q) vérifie si oui ou non le point q se trouve sur le cadre (lignes inclus et pas au centre vide). La méthode move(Point m1, Point m2) déplace la perle en déplaçant le point p selon le vecteur décrit par les deux points m1 et m2 .	
3.	La classe Frames possède une liste de cadres appelée alFrames . Les méthodes add(...) , get() , toArray() et clear(...) sont des méthodes déléguées de la liste. La méthode select(Point q) vérifie si un point q se trouve sur un des cadres de la liste. S'il y en a une ou plusieurs la méthode retourne d'indice de celui dessinée en dernier, sinon -1. Elle utilise la méthode isIn(...) de Frame . Programmez la méthode draw(...) qui dessine tous les cadres. Programmez la méthode calcSizes(int pc) qui lance un calcul de l'attribut size pour tous les cadres de la liste	
4.	Pour respecter le principe MVC, le DrawPanel possède un attribut de la classe frames comme le MainFrame aussi. Dans DrawPanel , un manipulateur setFrames(...) est appelé à partir du MainFrame pour faire pointer celui du DrawPanel sur celui du MainFrame . La méthode paintComponent efface le drawPanel et lance le dessin de l'attribut frames s'il n'est pas null .	
5.	Le constructeur du MainFrame crée un nouvel objet de la classe Frames . On ajoute un cadre par un clic gauche dans le drawPanel , puis en tirant avec la souris jusqu'à ce que la taille désirée soit atteinte. Les coordonnées du clic initial deviennent les coordonnées des point opposés du cadre. Comme, au début l'attribut size a la valeur 0, uniquement la ligne extérieure du cadre n'est dessinée. L'attribut frame de la classe Frame sert à mémoriser ce nouveau cadre.	
6.	Le bouton droit de la souris sert à en sélectionner un cadre pour le déplacer ensuite en tirant avec la souris (ayant le bouton droit appuyé). L'attribut mv sert à effectuer le déplacement du cadre sélectionné. Il est null dès qu'on clique avec la touche gauche. Il est ainsi possible de distinguer les deux actions possibles (ajout et déplacement). L'attribut frame de la classe Frame sert à mémoriser le cadre sélectionné par la souris.	

7.	Quand on lâche la souris après la création d'un nouveau cadre, alors l'attribut <i>size</i> de ce dernier est défini selon le pourcentage donné par le slider.	
8.	Le slider varie de 0 à 100, il est initialisé avec la valeur 30. Sa valeur est continuellement affichée dans un libellé au-dessous de lui. La modification de la valeur du slider, provoque un recalcul de l'attribut size pour chaque cadre de la liste.	

Schéma UML



Copie d'écran

