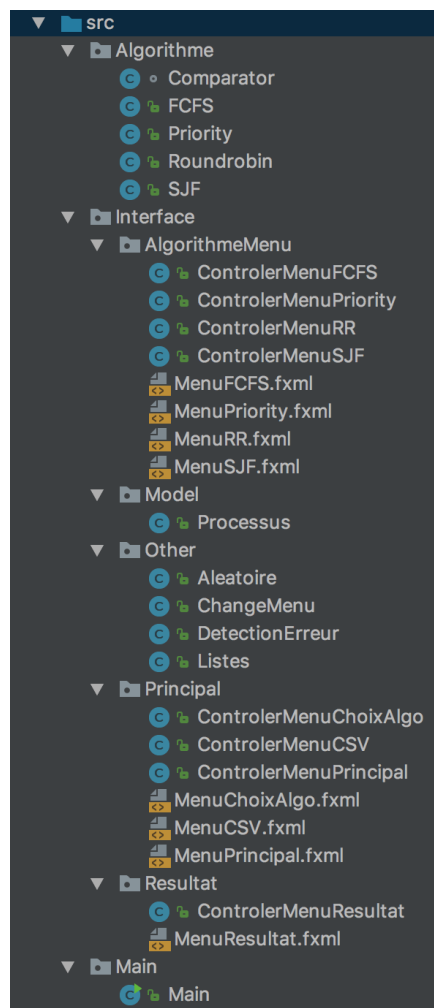


Les étapes de création du projet

1- Architecture Logiciel :

- Identification des principaux acteurs.
- Identification des principaux objets.
- Représentation des objets un utilisant le diagramme de classe UML.
- Détermination des relations entre les objets.
- Le projet est structurer de cette façon :

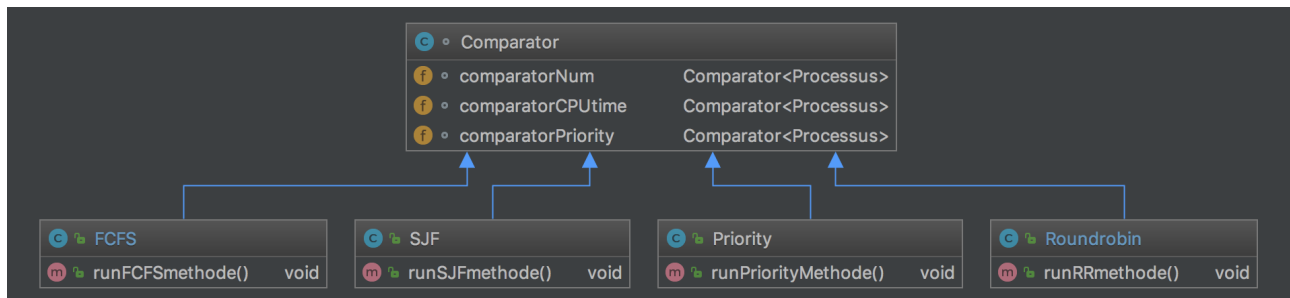


- le choix de cette structure reflète plusieurs critère mais principalement dans le point de vue conception orienté objet.

2- Création des classes :

Package Algorithmme :

- Dans ce package en retrouve pour chaque algorithmes de scheduling une classe spécifique a ce dernier avec toutes ces méthode de calculer ...etc.
- la classe Comparator est une class additionnel qui permet de trier une ArrayListe avec un critère précis.

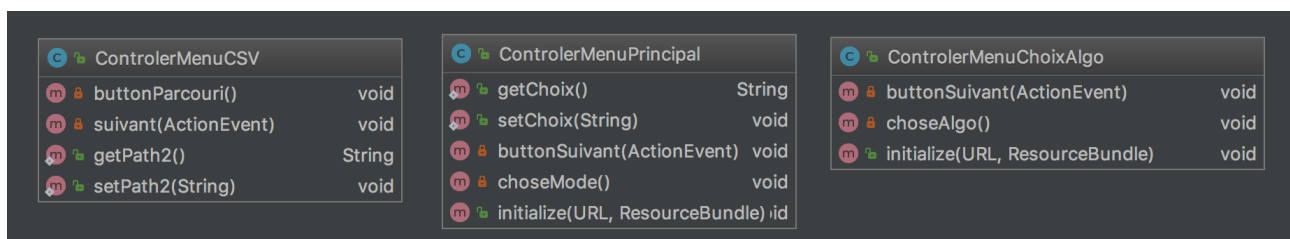


Package Interface :

- Dans ce package en retrouve tous les classe qui concerne l'interface du programme.
- ce package est diviser en 5 autre sous packages.

Package Principal :

- Dans ce package en retrouve les 3 menu principal en .fxml avec leur contrôler.
- Chaque contrôler de class a ces propre méthode de control de l'interface.



Package AlgorithmeMenu :

- Comme son nom l'indique dans ce package on retrouve les 4 menu de saisi de données pour chaque algorithme.
- Chaque contrôleur de class a ces propre méthode de control de l'interface.

C ControlerMenuFCFS m add() void m ajouterProcessus(ActionEvent) m calculerFCFS(ActionEvent) void m addProcessusAleatoirement() m addProcessusCSV() void m initialize(URL, ResourceBundle)	C ControlerMenuRR m getQuantum() int m add() void m ajouterProcessus(ActionEvent) void m calculerRR(ActionEvent) void m addProcessusAleatoirement() void m addProcessusCSV() void m initialize(URL, ResourceBundle) void	C ControlerMenuPriority m add() void m ajouterProcessus(ActionEvent) m calculerPriority(ActionEvent) void m addProcessusAleatoirement() m addProcessusCSV() void m initialize(URL, ResourceBundle)
C ControlerMenuSJF m add() void m ajouterProcessus(ActionEvent) m calculerSJF(ActionEvent) void m addProcessusAleatoirement() m addProcessusCSV() void m initialize(URL, ResourceBundle)		

Package Résultat :

- Dans ce package il y a le menu résultat avec son contrôleur

C ControlerMenuResultat
m retourMenuPrincipal(ActionEvent) void
m quiter(ActionEvent) void
m getChoix() String
m setChoix(String) void
m initialize(URL, ResourceBundle) void

Package Other :

- Dans ce package en retrouver des class complémentaire qui aide au bon fonctionnement du programme

<div><div>C</div><div>Listes</div><div><div>m</div><div>getAvg()</div><div>ArrayList<Float></div></div><div><div>m</div><div>getListProcessusesFCFS()</div><div>.ist<Processus></div></div><div><div>m</div><div>getListProcessusesSJF()</div><div>List<Processus></div></div><div><div>m</div><div>getListProcessusesPriority()</div><div>t<Processus></div></div><div><div>m</div><div>getListProcessusesRR()</div><div>leList<Processus></div></div></div>	<div><div>C</div><div>Aleatoire</div><div><div>m</div><div>getRandom()</div><div>Random</div></div><div><div>m</div><div>getNbrProcessus()</div><div>int</div></div><div><div>m</div><div>getCpuTime()</div><div>int</div></div><div><div>m</div><div>getPriority()</div><div>int</div></div></div>
<div><div>C</div><div>ChangeMenu</div><div><div>m</div><div>afficheMenuResultat(ActionEvent)</div><div>void</div></div><div><div>m</div><div>afficheMenuAlgo(ActionEvent, String, String)</div><div>void</div></div><div><div>m</div><div>afficheMenuChoixAlgo(ActionEvent)</div><div>void</div></div></div>	<div><div>C</div><div>DetectionErreur</div><div><div>m</div><div>isInputValid(TextField, String)</div><div>boolean</div></div><div><div>m</div><div>errorFichierCSV()</div><div>void</div></div></div>

Package Model :

- Dans ce package en retrouve notre model de processus.

<div><div>C</div><div>Processus</div><div><div>f</div><div>nom</div><div>StringProperty</div></div><div><div>f</div><div>num</div><div>IntegerProperty</div></div><div><div>f</div><div>cpuTime</div><div>IntegerProperty</div></div><div><div>f</div><div>priority</div><div>IntegerProperty</div></div><div><div>f</div><div>waitTime</div><div>FloatProperty</div></div><div><div>f</div><div>turnAroundTime</div><div>FloatProperty</div></div></div>

Puis le dernier package « Main » en retrouve notre main pour run le programme.

3- Fonctionnement de l'interface :

