

Contents

1	Package	2
1.1	Class DevOO	2
1.1.1	Declaration	2
1.1.2	Constructor summary	2
1.1.3	Method summary	2
1.1.4	Constructors	2
1.1.5	Methods	3

Class Hierarchy

Classes

- java.lang.Object
 - DevOO
 - controleur.Controleur
 - controleur.ControleurDonnees
 - java.lang.Enum
 - vue.ObjetVisualisable.CouleurTexte
 - javafx.application.Application
 - vue.FenetrePrincipale
 - javafx.scene.Node
 - javafx.scene.Parent
 - javafx.scene.layout.Region
 - javafx.scene.control.Control
 - javafx.scene.control.Labeled
 - javafx.scene.control.ButtonBase
 - javafx.scene.control.Button
 - vue.BoutonObservateur
 - javafx.scene.shape.Shape
 - javafx.scene.text.Text
 - vue.ObservateurMessageChamps
 - vue.ObjetVisualisable
 - vue.DetailFenetre
 - vue.DetailLivraison
 - vue.VueGraphiqueAideur
 - vue.VuePrincipale
 - vue.VueTextuelle

Interfaces

- controleur.ControleurInterface

Chapter 1

Package controleur

<i>Package Contents</i>	<i>Page</i>
Interfaces	
ControleurInterface ??	
Interface le controleur	
Classes	
Controleur ??	
Implmente l'interface controleur.	
ControleurDonnees ??	
Cette classe contient les donnees necessaires pour la gestion des tats.	

1.1 Interface ControleurInterface

Interface le controleur

1.1.1 Declaration

```
public interface ControleurInterface
```

1.1.2 All known subinterfaces

Controleur

1.1.3 All classes known to implement interface

Controleur

1.1.4 Method summary

[ajouterActivationFonctionnalitesObservateur\(ActivationFonctionnalitesObservateur\)](#)

Ajoute un observateur pour l'activation des fonctionnalites

principales de l'application

ajouterActivationOuvrirDemandeObservateur(ActivationOuvrirDemandeObservateur)

Ajoute un observateur des changement du plan

ajouterActivationOuvrirPlanObservateur(ActivationOuvrirPlanObservateur)

Ajoute un observateur du chargement du plan

ajouterAnnulerCommandeObservateur(AnnulerCommandeObservateur)

Ajoute un observateur l'annulation d'une commande

ajouterMessageObservateur(MessageObservateur)

Ajoute un observateur des messages envoys

ajouterModeleObservateur(ModeleObservateur) Ajoute

un observateur au changement du modle

ajouterPlanChargeObserveur(PlanChargeObservateur)

ajouterRetablirCommandeObservateur(RetablirCommandeObservateur)

Ajoute un observateur au rtablissement d'une commande

ajouterTourneeObservateur(ActivationFonctionnalitesObservateur)

Ajoute un observateur la tourne

chargerLivraisons(File) Cette methode essaye de convertir un fichier XML dans sa representation d'objets.

chargerPlan(File) Cette methode essaye de convertir un fichier XML dans sa representation d'objets.

clicAnnuler() Appel lors d'un clic sur Annuler

clicCalculTournee() Appel lors du clic sur le calcul de la tourne

clicDroit() Appel lors du clic droit

clicOutilAjouter() Appel lors du clic pour passer dans le mode d'ajout

clicOutilEchanger() Appel lors du clic pour passer dans le mode d'change

clicOutilSupprimer() Appel lors du clic pour passer dans le mode de suppression

clicRetablir() Appel lors d'un clic sur Rtablir

clicSurLivraison(int) Appel quand il y a un clic sur une livraison

clicSurPlan(int) Appel quand il y a un clic sur plan

genererFeuilleDeRoute(File) Gñre la feuille de route

getModele() Retourne le modle

getPlanDeVille() Retourne le plan de la ville

1.1.5 Methods

- **ajouterActivationFonctionnalitesObservateur**

```
void ajouterActivationFonctionnalitesObservateur(  
    observateur.
```

ActivationFonctionnalitesObservateur observateur)

– **Description**

Ajoute un observateur pour l'activation des fonctionnalits principales de l'application

– **Parameters**

* observateur –

• **ajouterActivationOuvrirDemandeObservateur**

```
void ajouterActivationOuvrirDemandeObservateur(  
    observateur.ActivationOuvrirDemandeObservateur  
    planObserveur)
```

– **Description**

Ajoute un observateur des changement du plan

– **Parameters**

* planObserveur –

• **ajouterActivationOuvrirPlanObservateur**

```
void ajouterActivationOuvrirPlanObservateur(  
    observateur.ActivationOuvrirPlanObservateur  
    chargementPlanObserveur)
```

– **Description**

Ajoute un observateur du chargement du plan

– **Parameters**

* chargementPlanObserveur –

• **ajouterAnnulerCommandeObservateur**

```
void ajouterAnnulerCommandeObservateur(observateur  
    .AnnulerCommandeObservateur  
    annulerCommandeObserver)
```

– **Description**

Ajoute un observateur l'annulation d'une commande

– **Parameters**

* annulerCommandeObservateur –

- **ajouterMessageObservateur**

```
void ajouterMessageObservateur(observateur.  
    MessageObservateur obs)
```

- **Description**

Ajoute un observateur des messages envoys

- **Parameters**

* obs –

- **ajouterModeleObservateur**

```
void ajouterModeleObservateur(observateur.  
    ModeleObservateur observeur)
```

- **Description**

Ajoute un observateur au changement du modle

- **Parameters**

* observeur –

- **ajouterPlanChargeObservateur**

```
void ajouterPlanChargeObservateur(observateur.  
    PlanChargeObservateur planChargeObservateur)
```

- **ajouterRetablirCommandeObservateur**

```
void ajouterRetablirCommandeObservateur(  
    observateur.RetablirCommandeObservateur  
    retablirCommandeObservateur)
```

- **Description**

Ajoute un observateur au rtablissement d'une commande

- **Parameters**

* retablirCommandeObservateur –

- **ajouterTourneeObservateur**

```
void ajouterTourneeObservateur(observateur.  
    ActivationFonctionnalitesObservateur  
    tourneeObserveur)
```

– **Description**

Ajoute un observateur la tourne

– **Parameters**

* `tourneeObserveur` –

• **chargerLivraisons**

```
void chargerLivraisons(java.io.File  
    fichierLivraisons) throws java.lang.Exception
```

– **Description**

Cette methode essaye de convertir un fichier XML dans sa representation d'objets.

– **Parameters**

* `fichierLivraisons` – Objet File qui represente le fichier XML

– **Throws**

* `java.lang.Exception` – Lance une exception s'il y a une erreur lors du chargement des objets

• **chargerPlan**

```
void chargerPlan(java.io.File fichierPlan) throws  
    java.lang.Exception
```

– **Description**

Cette methode essaye de convertir un fichier XML dans sa representation d'objets.

– **Parameters**

* `fichierPlan` – Objet File qui represente le fichier XML

– **Throws**

* `java.lang.Exception` – Lance une exception s'il y a une erreur lors du chargement des objets

• **clicAnnuler**

void clicAnnuler()

– **Description**

Appel lors d'un clic sur Annuler

• **clicCalculTournee**

void clicCalculTournee()

– **Description**

Appel lors du clic sur le calcul de la tourne

• **clicDroit**

void clicDroit()

– **Description**

Appel lors du clic droit

• **clicOutilAjouter**

void clicOutilAjouter()

– **Description**

Appel lors du clic pour passer dans le mode d'ajout

• **clicOutilEchanger**

void clicOutilEchanger()

– **Description**

Appel lors du clic pour passer dans le mode d'change

• **clicOutilSupprimer**

void clicOutilSupprimer()

– **Description**

Appel lors du clic pour passer dans le mode de suppression

- **clicRetablir**

void clicRetablir()

- **Description**

Appel lors d'un clic sur Rtablir

- **clicSurLivraison**

void clicSurLivraison(int livraisonId)

- **Description**

Appel quand il y a un clic sur une livraison

- **Parameters**

* **livraisonId** – L'identifiant de la livraison

- **clicSurPlan**

void clicSurPlan(int intersectionId)

- **Description**

Appel quand il y a un clic sur plan

- **Parameters**

* **intersectionId** – L'identifiant de l'intersection cliqu

- **genererFeuilleDeRoute**

void genererFeuilleDeRoute(java.io.File fichier)
throws controleur.commande.CommandeException

- **Description**

Gnre la feuille de route

- **Parameters**

* **fichier** – Le fichier dans lequel on devra crire la feuille de route

- **Throws**

* **controleur.commande.CommandeException** – Une erreur lors de l'excution de la commande de gnration

- **getModele**

```
modele.donneesxml.ModeleLecture getModele()
```

- **Description**

Retourn le modle

- **Returns** – Le modle actuel en lecture

- **getPlanDeVille**

```
modele.donneesxml.PlanDeVille getPlanDeVille()
```

- **Description**

Retourne le plan de la ville

- **Returns** – Le plan de la ville actuellement charg

1.2 Class Controleur

Implmente l'interface controleur. Point d'entre principal pour toutes les interactions avec le package vue.

1.2.1 Declaration

```
public class Controleur
    extends java.lang.Object implements
        ControleurInterface
```

1.2.2 Constructor summary

[Controleur\(\)](#) Constructeur public du controleur

1.2.3 Method summary

[ajouterActivationFonctionnalitesObservateur\(ActivationFonctionnalitesObservateur\)](#)
[ajouterActivationOuvrirDemandeObservateur\(ActivationOuvrirDemandeObservateur\)](#)
[ajouterActivationOuvrirPlanObservateur\(ActivationOuvrirPlanObservateur\)](#)
[ajouterAnnulerCommandeObservateur\(AnnulerCommandeObservateur\)](#)
[ajouterMessageObservateur\(MessageObservateur\)](#)
[ajouterModeleObservateur\(ModeleObservateur\)](#)
[ajouterPlanChargeObserveur\(PlanChargeObservateur\)](#)
[ajouterRetablirCommandeObservateur\(RetablirCommandeObservateur\)](#)
[ajouterTourneeObservateur\(ActivationFonctionnalitesObservateur\)](#)

```

chargerLivraisons(File)
chargerPlan(File)
clicAnnuler()
clicCalculTournee()
clicDroit()
clicOutilAjouter()
clicOutilEchanger()
clicOutilSupprimer()
clicRetablir()
clicSurLivraison(int)
clicSurPlan(int)
genererFeuilleDeRoute(File)
getModele()
getPlanDeVille()

```

1.2.4 Constructors

- **Controleur**

```
public Controleur()
```

- **Description**

Constructeur public du controleur

1.2.5 Methods

- **ajouterActivationFonctionnalitesObservateur**

```
void ajouterActivationFonctionnalitesObservateur(
    observateur.
    ActivationFonctionnalitesObservateur observeur)

```

- **Description copied from [ControleurInterface](#)**

Ajoute un observateur pour l'activation des fonctionnalits principales de l'application

- **Parameters**

* `observeur` –

- **ajouterActivationOuvrirDemandeObservateur**

```
void ajouterActivationOuvrirDemandeObservateur(
    observateur. ActivationOuvrirDemandeObservateur
    planObserveur)

```

- **Description copied from [ControleurInterface](#)**

Ajoute un observateur des changement du plan

- **Parameters**

* planObserveur –

- **ajouterActivationOuvrirPlanObservateur**

```
void ajouterActivationOuvrirPlanObservateur (
    observateur . ActivationOuvrirPlanObservateur
    chargementPlanObserveur)
```

- **Description copied from [ControleurInterface](#)**

Ajoute un observateur du chargement du plan

- **Parameters**

* chargementPlanObserveur –

- **ajouterAnnulerCommandeObservateur**

```
void ajouterAnnulerCommandeObservateur ( observateur
    . AnnulerCommandeObservateur
    annulerCommandeObserveur )
```

- **Description copied from [ControleurInterface](#)**

Ajoute un observateur l'annulation d'une commande

- **Parameters**

* annulerCommandeObserveur –

- **ajouterMessageObservateur**

```
void ajouterMessageObservateur ( observateur .
    MessageObservateur obs )
```

- **Description copied from [ControleurInterface](#)**

Ajoute un observateur des messages envoys

- **Parameters**

* obs –

- **ajouterModeleObservateur**

void ajouterModeleObservateur(observateur.
ModeleObservateur observeur)

– **Description copied from [ControleurInterface](#)**

Ajoute un observateur au changement du modle

– **Parameters**

* observeur –

- **ajouterPlanChargeObservateur**

void ajouterPlanChargeObservateur(observateur.
PlanChargeObservateur planChargeObservateur)

- **ajouterRetablirCommandeObservateur**

void ajouterRetablirCommandeObservateur(
observateur.RetablirCommandeObservateur
retablirCommandeObserver)

– **Description copied from [ControleurInterface](#)**

Ajoute un observateur au rtablissement d’une commande

– **Parameters**

* retablirCommandeObserver –

- **ajouterTournéeObservateur**

void ajouterTournéeObservateur(observateur.
ActivationFonctionnalitesObservateur
tourneeObserver)

– **Description copied from [ControleurInterface](#)**

Ajoute un observateur la tourne

– **Parameters**

* tourneeObserver –

- **chargerLivraisons**

void chargerLivraisons(java.io.File
fichierLivraisons) **throws** java.lang.Exception

- **Description copied from [ControleurInterface](#)**
Cette methode essaye de convertir un fichier XML dans sa representation d'objets.
- **Parameters**
 - * `fichierLivraisons` – Objet File qui represente le fichier XML
- **Throws**
 - * `java.lang.Exception` – Lance une exception s'il y a une erreur lors du chargement des objets

- **chargerPlan**

```
void chargerPlan(java.io.File fichierPlan) throws
    java.lang.Exception
```

- **Description copied from [ControleurInterface](#)**
Cette methode essaye de convertir un fichier XML dans sa representation d'objets.
- **Parameters**
 - * `fichierPlan` – Objet File qui represente le fichier XML
- **Throws**
 - * `java.lang.Exception` – Lance une exception s'il y a une erreur lors du chargement des objets

- **clicAnnuler**

```
void clicAnnuler()
```

- **Description copied from [ControleurInterface](#)**
Appel lors d'un clic sur Annuler

- **clicCalculTournee**

```
void clicCalculTournee()
```

- **Description copied from [ControleurInterface](#)**
Appel lors du clic sur le calcul de la tourne

- **clicDroit**

void clicDroit()

- **Description copied from [ControleurInterface](#)**
Appel lors du clic droit

- **clicOutilAjouter**

void clicOutilAjouter()

- **Description copied from [ControleurInterface](#)**
Appel lors du clic pour passer dans le mode d'ajout

- **clicOutilEchanger**

void clicOutilEchanger()

- **Description copied from [ControleurInterface](#)**
Appel lors du clic pour passer dans le mode d'change

- **clicOutilSupprimer**

void clicOutilSupprimer()

- **Description copied from [ControleurInterface](#)**
Appel lors du clic pour passer dans le mode de suppression

- **clicRetablir**

void clicRetablir()

- **Description copied from [ControleurInterface](#)**
Appel lors d'un clic sur Rtablir

- **clicSurLivraison**

void clicSurLivraison(int livraisonId)

- **Description copied from [ControleurInterface](#)**
Appel quand il y a un clic sur une livraison
- **Parameters**

* livraisonId – L’identifiant de la livraison

- clicSurPlan

void clicSurPlan(**int** intersectionId)

- **Description copied from [ControleurInterface](#)**

Appel quand il y a un clic sur plan

- **Parameters**

* intersectionId – L’identifiant de l’intersection cliqu

- genererFeuilleDeRoute

void genererFeuilleDeRoute(**java.io.File** fichier)
throws controleur.commande.CommandeException

- **Description copied from [ControleurInterface](#)**

Gnre la feuille de route

- **Parameters**

* fichier – Le fichier dans lequel on devra crire la feuille de route

- **Throws**

* controleur.commande.CommandeException – Une erreur lors de l’excution de la commande de gnration

- getModele

modele.donneesxml.ModeleLecture getModele()

- **Description copied from [ControleurInterface](#)**

Retourn le modle

- **Returns** – Le modle actuel en lecture

- getPlanDeVille

modele.donneesxml.PlanDeVille getPlanDeVille()

- **Description copied from [ControleurInterface](#)**

Retourne le plan de la ville

- **Returns** – Le plan de la ville actuellement charg

1.3 Class ControleurDonnees

Cette classe contient les donnees necessaires pour la gestion des tats. On pourrait dire qu'elle represente seulement des donnees et devrait du coup mieux etre situee dans le package modele. Par contre elle est liee a une seule IHM, du coup il y a une bonne raison pour la laisser ici dans le controleur.

1.3.1 Declaration

```
public class ControleurDonnees
    extends java.lang.Object
```

1.3.2 Constructor summary

[ControleurDonnees\(\)](#)

1.3.3 Method summary

[ajouterActivationObservateur\(ActivationFonctionnalitesObservateur\)](#)

Ajoute un observateur d'activation

[ajouterAnnulerCommandeObservateur\(AnnulerCommandeObservateur\)](#)

Ajoute un observateur de la commande annuler

[ajouterChargementPlanObservateur\(ActivationOuvrirPlanObservateur\)](#)

Ajoute un observateur du chargement du plan

[ajouterCommande\(Commande\)](#) Ajoute une commande
l'historique

[ajouterModeleObservateur\(ModeleObservateur\)](#) Ajoute
un observateur du modele

[ajouterPlanChargeObservateur\(PlanChargeObservateur\)](#)

[ajouterPlanObservateur\(ActivationOuvrirDemandeObservateur\)](#)

[ajouterRetablirCommandeObservateur\(RetablirCommandeObservateur\)](#)

Ajoute un observateur de la commande retablir

[effacerCommandeAAnnuler\(\)](#) Efface la liste des commandes
annuler et notifie la vue qu'il doit desactiver l'element du menu
correspondant

[effacerCommandesARetablir\(\)](#) Efface la liste des commandes
retablir et notifie la vue qu'elle doit dsactiver l'element du
menu correspondant

[effacerHistorique\(\)](#) Efface l'historique (vide les commandes
annulable et rtablissable(?))

[getHist\(\)](#)

[getModele\(\)](#) Retourne le modele associ

[getPlan\(\)](#) Retourne le plan de la ville

[notifierObservateurOuvrirDemande\(boolean\)](#) Notifie les
observateurs du plan

notifierObservateurOuvrirPlan(boolean) Notifie les observateurs du chargement du plan

notifierObservateursActivation(boolean) Notifie les observateurs de l'activation

notifierObservateursAnnuler(boolean) Notifie les observateurs qu'il y a eu une annulation

notifierObservateursCalculTournee(boolean) Notifie les observateurs du calcul de la tournée

notifierObservateursMessage(String) Notifie les observateurs qu'il y a un message

notifierObservateursModele() Notifie les observateurs du changement du modèle

notifierObservateursRetablir(boolean) Notifie les observateurs qu'il y a eu un rétablissement

notifierPlanChargeObservateur()

setHistorique(Historique) Affecte l'historique

setModele(Modele) Affecte le modèle

setPlan(PlanDeVille) Affecte le plan de la ville

1.3.4 Constructors

- **ControleurDonnees**

```
public ControleurDonnees()
```

1.3.5 Methods

- **ajouterActivationObservateur**

```
public void ajouterActivationObservateur(
    observateur.
    ActivationFonctionnalitesObservateur obs)
```

– Description

Ajoute un observateur d'activation

– Parameters

* **obs** – L'objet observateur

- **ajouterAnnulerCommandeObservateur**

```
public void ajouterAnnulerCommandeObservateur(
    observateur. AnnulerCommandeObservateur obs)
```

- **Description**
Ajoute un observateur de la commande annuler
- **Parameters**
 - * **obs** – L’objet observateur

- **ajouterChargementPlanObservateur**

```
public void ajouterChargementPlanObservateur(
    observateur.ActivationOuvrirPlanObservateur
    chargementPlanObserveur)
```

- **Description**
Ajoute un observateur du chargement du plan
- **Parameters**
 - * **chargementPlanObserveur** – L’objet observateur

- **ajouterCommande**

```
public void ajouterCommande(commande.Commande
    commande)
```

- **Description**
Ajoute une commande l’historique
- **Parameters**
 - * **commande** – Une commande excute

- **ajouterModeleObservateur**

```
public void ajouterModeleObservateur(observateur.
    ModeleObservateur obs)
```

- **Description**
Ajoute un observateur du modle
- **Parameters**
 - * **obs** – L’objet observateur

- **ajouterPlanChargeObservateur**

```
public void ajouterPlanChargeObservateur(  
    observateur.PlanChargeObservateur  
    planChargeObservateur)
```

- **ajouterPlanObservateur**

```
public void ajouterPlanObservateur(observateur.  
    ActivationOuvrirDemandeObservateur  
    planObserver)
```

- **ajouterRetablirCommandeObservateur**

```
public void ajouterRetablirCommandeObservateur(  
    observateur.RetablirCommandeObservateur obs)
```

- **Description**

Ajoute un observateur de la commande rtablir

- **Parameters**

* **obs** – L’objet observateur

- **effacerCommandeAAnnuler**

```
public void effacerCommandeAAnnuler()
```

- **Description**

Efface la liste des commandes annuler et notifie la vue qu’il doit dsactiver l’lment du menu correspondant

- **effacerCommandesARetablir**

```
public void effacerCommandesARetablir()
```

- **Description**

Efface la liste des commandes retablir et notifie la vue qu’elle doit dsactiver l’lment du menu correspondant

- **effacerHistorique**

```
public void effacerHistorique()
```

- **Description**

Efface l'historique (vide les commandes annulable et rtablissable(?))

- **getHist**

```
public commande.Historique getHist()
```

- **getModele**

```
public modele.donneesxml.Modele getModele()
```

- **Description**

Retourne le modle associ

- **Returns** – Le modle associ

- **getPlan**

```
public modele.donneesxml.PlanDeVille getPlan()
```

- **Description**

Retourne le plan de la ville

- **Returns** – Le plan de la ville

- **notifierObservateurOuvrirDemande**

```
public void notifierObservateurOuvrirDemande(
    boolean activer)
```

- **Description**

Notifie les observateurs du plan

- **Parameters**

- * **activer** – Vrai s'il faut envoyer un message d'activation aux observateurs

- **notifierObservateurOuvrirPlan**

```
public void notifierObservateurOuvrirPlan(boolean
    activer)
```

- **Description**

Notifie les observateurs du chargement du plan

- **Parameters**

- * **activer** – Vrai s'il faut envoyer un message d'activation aux observateurs

- **notifierObservateursActivation**

```
public void notifierObservateursActivation(boolean
    etat)
```

- **Description**

Notifie les observateurs de l'activation

- **Parameters**

- * **etat** – Vrai s'il faut activer les observateurs

- **notifierObservateursAnnuler**

```
public void notifierObservateursAnnuler(boolean
    activation)
```

- **Description**

Notifie les observateurs qu'il y a eu une annulation

- **Parameters**

- * **activation** – Vrai si les observateurs doivent s'activer dans ce cas d'annulation

- **notifierObservateursCalculTournee**

```
public void notifierObservateursCalculTournee(
    boolean activation)
```

- **Description**

Notifie les observateurs du calcul de la tournée

- **Parameters**

- * **activation** – Vrai si les observateurs doivent s'activer

- **notifierObservateursMessage**

```
public void notifierObservateursMessage(java.lang.  
String message)
```

– **Description**

Notifie les observateurs qu'il y a un message

– **Parameters**

* **message** – Le message envoy

• **notifierObservateursModele**

```
public void notifierObservateursModele()
```

– **Description**

Notifie les observateurs du changemetn du modle

• **notifierObservateursRetablir**

```
public void notifierObservateursRetablir(boolean  
activation)
```

– **Description**

Notifie les observateurs qu'il y eu un rtablissement

– **Parameters**

* **activation** – Vrai si les observateurs doivent s'activer dans ce cas de rtablissement

• **notifierPlanChargeObservateur**

```
public void notifierPlanChargeObservateur()
```

• **setHistorique**

```
public void setHistorique(commande.Historique hist  
)
```

– **Description**

Affecte l'historique

– **Parameters**

* **hist** – Le nouvel historique

- **setModele**

```
public void setModele(modele.donneesxml.Modele  
    modele)
```

- **Description**

Affecte le modle

- **Parameters**

- * **modele** – Le nouveau modle

- **setPlan**

```
public void setPlan(modele.donneesxml.PlanDeVille  
    plan)
```

- **Description**

Affecte le plan de la ville

- **Parameters**

- * **plan** – Le nouveau plan de la ville

Chapter 2

Package modele

Package Contents

Page

Chapter 3

Package vue

<i>Package Contents</i>	<i>Page</i>
Classes	
BoutonObservateur	??
Bouton particulier qui observe les modifications au niveau du modle pour savoir s'il doit s'activer ou pas.	
DetailFenetre	??
Gre l'affichage sous forme textuelle des details d'une fentre de livraison dans la TreeTableView.	
DetailLivraison	??
Gre l'affichage sous forme textuelle des details d'une livraison dans la TreeTableView.	
FenetrePrincipale	??
Cette classe cre la fenetre principale avec ses enfants.	
ObjetVisualisable	??
Cette classe permet de visualiser une fentre (de livraison) ou une livraison sous forme textuelle.	
ObjetVisualisable.CouleurTexte	??
Diffrentes couleurs possibles pour un lment dans la liste.	
ObserveurMessageChamps	??
Champ texte, couteur des messages qui peuvent tre reu	
VueGraphiqueAideur	??
Contient les mthodes permettant d'afficher les lments dans la zone graphique	
VuePrincipale	??
Cette classe joue le rle de binding pour la fenetre principale de l'application.	
VueTextuelle	??
Cette classe gre les livraisons et leurs horaires.	

Ce package contient les diffrentes classes permettant l'interaction avec la vue. Elle se charge de crer le controleur et lancer la fentre principale de l'application.

3.1 Class BoutonObservateur

Bouton particulier qui observe les modifications au niveau du modle pour savoir s'il doit s'activer ou pas.

3.1.1 Declaration

```
public class BoutonObservateur
    extends javafx.scene.control.Button implements
        controleur.observeur.
        ActivationFonctionnalitesObservateur
```

3.1.2 Constructor summary

[BoutonObservateur\(\)](#)

3.1.3 Method summary

[notifierObservateursActivation\(boolean\)](#)

3.1.4 Constructors

- BoutonObservateur

```
public BoutonObservateur()
```

3.1.5 Methods

- notifierObservateursActivation

```
void notifierObservateursActivation(boolean
    desactiver)
```

– Description copied from [controleur.observeur.ActivationFonctionnalitesObservateur](#)

Notifie les observateurs qui attendent un message d'activation notamment les boutons de fonctionnalits (ajouter, supprimer, echanger) qu'il faut changer l'tat d'activation

– Parameters

* **desactiver** – Vrai si on doit dsactiver les observeurs

3.1.6 Members inherited from class Button

`javafx.scene.control.Button`

- `public final BooleanProperty cancelButtonProperty()`
- `protected Skin createDefaultSkin()`
- `public final BooleanProperty defaultButtonProperty()`
- `public void fire()`
- `public final boolean isCancelButton()`
- `public final boolean isDefaultButton()`
- `public final void setCancelButton(boolean arg0)`
- `public final void setDefaultButton(boolean arg0)`

3.1.7 Members inherited from class ButtonBase

`javafx.scene.control.ButtonBase`

- `public void arm()`
- `public final ReadOnlyBooleanProperty armedProperty()`
- `public void disarm()`
- `public void executeAccessibleAction(javafx.scene.AccessibleAction arg0, java.lang.Object[] arg1)`
- `public abstract void fire()`
- `public final EventHandler getOnAction()`
- `public final boolean isArmed()`
- `public final ObjectProperty onActionProperty()`
- `public final void setOnAction(javafx.event.EventHandler arg0)`

3.1.8 Members inherited from class Labeled

`javafx.scene.control.Labeled`

- `public final ObjectProperty alignmentProperty()`
- `public final ObjectProperty contentDisplayProperty()`
- `public final StringProperty ellipsisStringProperty()`
- `public final ObjectProperty fontProperty()`
- `public final Pos getAlignment()`
- `public static List getClassCssMetaData()`
- `public Orientation getContentBias()`
- `public final ContentDisplay getContentDisplay()`
- `public List getControlCssMetaData()`
- `public final String getEllipsisString()`
- `public final Font getFont()`
- `public final Node getGraphic()`
- `public final double getGraphicTextGap()`
- `public final Insets getLabelPadding()`
- `public final double getLineSpacing()`
- `public final String getText()`
- `public final TextAlignment getTextAlignment()`
- `public final Paint getTextFill()`
- `public final OverrunStyle getTextOverrun()`
- `public final ObjectProperty graphicProperty()`
- `public final DoubleProperty graphicTextGapProperty()`
- `protected Pos impl_cssGetAlignmentInitialValue()`
- `public final boolean isMnemonicParsing()`

- public final boolean isUnderline()
- public final boolean isWrapText()
- public final ReadOnlyObjectProperty labelPaddingProperty()
- public final DoubleProperty lineSpacingProperty()
- public final BooleanProperty mnemonicParsingProperty()
- public final void setAlignment(javafx.geometry.Pos arg0)
- public final void setContentDisplay(ContentDisplay arg0)
- public final void setEllipsisString(java.lang.String arg0)
- public final void setFont(javafx.scene.text.Font arg0)
- public final void setGraphic(javafx.scene.Node arg0)
- public final void setGraphicTextGap(double arg0)
- public final void setLineSpacing(double arg0)
- public final void setMnemonicParsing(boolean arg0)
- public final void setText(java.lang.String arg0)
- public final void setTextAlignment(javafx.scene.text.TextAlignment arg0)
- public final void setTextFill(javafx.scene.paint.Paint arg0)
- public final void setTextOverrun(OverrunStyle arg0)
- public final void setUnderline(boolean arg0)
- public final void setWrapText(boolean arg0)
- public final ObjectProperty textAlignmentProperty()
- public final ObjectProperty textFillProperty()
- public final ObjectProperty textOverrunProperty()
- public final StringProperty textProperty()
- public String toString()
- public final BooleanProperty underlineProperty()
- public final BooleanProperty wrapTextProperty()

3.1.9 Members inherited from class Control

`javafx.scene.control.Control`

- protected double computeMaxHeight(double arg0)
- protected double computeMaxWidth(double arg0)
- protected double computeMinHeight(double arg0)
- protected double computeMinWidth(double arg0)
- protected double computePrefHeight(double arg0)
- protected double computePrefWidth(double arg0)
- public final ObjectProperty contextMenuProperty()
- protected Skin createDefaultSkin()
- public void executeAccessibleAction(javafx.scene.AccessibleAction arg0, java.lang.Object[] arg1)
- public double getBaselineOffset()
- public static List getClassCssMetaData()
- public final ContextMenu getContextMenu()
- protected List getControlCssMetaData()
- public final List getCssMetaData()
- public final Skin getSkin()
- public final Tooltip getTooltip()
- protected Boolean impl_cssGetFocusTraversableInitialValue()
- protected void impl_processCSS(javafx.beans.value.WritableValue arg0)
- public boolean isResizable()

- protected void `layoutChildren()`
- public Object `queryAccessibleAttribute(javafx.scene.AccessibleAttribute arg0, java.lang.Object[] arg1)`
- public final void `setContextMenu(ContextMenu arg0)`
- public final void `setSkin(Skin arg0)`
- public final void `setTooltip(Tooltip arg0)`
- protected StringProperty `skinClassNameProperty()`
- public final ObjectProperty `skinProperty()`
- public final ObjectProperty `tooltipProperty()`

3.1.10 Members inherited from class Region

`javafx.scene.layout.Region`

- public final ObjectProperty `backgroundProperty()`
- public final ObjectProperty `borderProperty()`
- public final BooleanProperty `cacheShapeProperty()`
- public final BooleanProperty `centerShapeProperty()`
- protected double `computeMaxHeight(double arg0)`
- protected double `computeMaxWidth(double arg0)`
- protected double `computeMinHeight(double arg0)`
- protected double `computeMinWidth(double arg0)`
- protected double `computePrefHeight(double arg0)`
- protected double `computePrefWidth(double arg0)`
- public final Background `getBackground()`
- public final Border `getBorder()`
- public static List `getClassCssMetaData()`
- public List `getCssMetaData()`
- public final double `getHeight()`
- public final Insets `getInsets()`
- public final double `getMaxHeight()`
- public final double `getMaxWidth()`
- public final double `getMinHeight()`
- public final double `getMinWidth()`
- public final Insets `getOpaqueInsets()`
- public final Insets `getPadding()`
- public final double `getPrefHeight()`
- public final double `getPrefWidth()`
- public final Shape `getShape()`
- public String `getUserAgentStylesheet()`
- public final double `getWidth()`
- public final ReadOnlyDoubleProperty `heightProperty()`
- protected boolean `impl_computeContains(double arg0, double arg1)`
- public BaseBounds `impl_computeGeomBounds(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)`
- protected final Bounds `impl_computeLayoutBounds()`
- public NGNode `impl_createPeer()`
- protected final void `impl_notifyLayoutBoundsChanged()`
- protected void `impl_pickNodeLocal(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- public void `impl_updatePeer()`
- public final ReadOnlyObjectProperty `insetsProperty()`

- `public final boolean isCacheShape()`
- `public final boolean isCenterShape()`
- `public boolean isResizable()`
- `public final boolean isScaleShape()`
- `public final boolean isSnapToPixel()`
- `protected void layoutInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.HPos arg6, javafx.geometry.VPos arg7)`
- `protected void layoutInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.Insets arg6, boolean arg7, boolean arg8, javafx.geometry.HPos arg9, javafx.geometry.VPos arg10)`
- `public static void layoutInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.Insets arg6, boolean arg7, boolean arg8, javafx.geometry.HPos arg9, javafx.geometry.VPos arg10, boolean arg11)`
- `protected void layoutInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.Insets arg6, javafx.geometry.HPos arg7, javafx.geometry.VPos arg8)`
- `public final double maxHeight(double arg0)`
- `public final DoubleProperty maxHeightProperty()`
- `public final double maxWidth(double arg0)`
- `public final DoubleProperty maxWidthProperty()`
- `public final double minHeight(double arg0)`
- `public final DoubleProperty minHeightProperty()`
- `public final double minWidth(double arg0)`
- `public final DoubleProperty minWidthProperty()`
- `public final ObjectProperty opaqueInsetsProperty()`
- `public final ObjectProperty paddingProperty()`
- `protected void positionInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.HPos arg6, javafx.geometry.VPos arg7)`
- `public static void positionInArea(javafx.scene.Node arg0, double arg1, double arg2, double arg3, double arg4, double arg5, javafx.geometry.Insets arg6, javafx.geometry.HPos arg7, javafx.geometry.VPos arg8, boolean arg9)`
- `public final double prefHeight(double arg0)`
- `public final DoubleProperty prefHeightProperty()`
- `public final double prefWidth(double arg0)`
- `public final DoubleProperty prefWidthProperty()`
- `public void resize(double arg0, double arg1)`
- `public final BooleanProperty scaleShapeProperty()`
- `public final void setBackground(Background arg0)`
- `public final void setBorder(Border arg0)`
- `public final void setCacheShape(boolean arg0)`
- `public final void setCenterShape(boolean arg0)`
- `protected void setHeight(double arg0)`
- `public final void setMaxHeight(double arg0)`
- `public void setMaxSize(double arg0, double arg1)`
- `public final void setMaxWidth(double arg0)`

- `public final void setMinHeight(double arg0)`
- `public void setMinSize(double arg0, double arg1)`
- `public final void setMinWidth(double arg0)`
- `public final void setOpaqueInsets(javafx.geometry.Insets arg0)`
- `public final void setPadding(javafx.geometry.Insets arg0)`
- `public final void setPrefHeight(double arg0)`
- `public void setPrefSize(double arg0, double arg1)`
- `public final void setPrefWidth(double arg0)`
- `public final void setScaleShape(boolean arg0)`
- `public final void setShape(javafx.scene.shape.Shape arg0)`
- `public final void setSnapToPixel(boolean arg0)`
- `protected void setWidth(double arg0)`
- `public final ObjectProperty shapeProperty()`
- `public final double snappedBottomInset()`
- `public final double snappedLeftInset()`
- `public final double snappedRightInset()`
- `public final double snappedTopInset()`
- `protected double snapPosition(double arg0)`
- `protected double snapSize(double arg0)`
- `protected double snapSpace(double arg0)`
- `public final BooleanProperty snapToPixelProperty()`
- `public static final USE_COMPUTED_SIZE`
- `public static final USE_PREF_SIZE`
- `public final ReadOnlyDoubleProperty widthProperty()`

3.1.11 Members inherited from class Parent

`javafx.scene.Parent`

- `protected double computeMinHeight(double arg0)`
- `protected double computeMinWidth(double arg0)`
- `protected double computePrefHeight(double arg0)`
- `protected double computePrefWidth(double arg0)`
- `public double getBaselineOffset()`
- `protected ObservableList getChildren()`
- `public ObservableList getChildrenUnmodifiable()`
- `public final ParentTraversalEngine getImpl_traversalEngine()`
- `protected List getManagedChildren()`
- `public final ObservableList getStylesheets()`
- `protected boolean impl_computeContains(double arg0, double arg1)`
- `public BaseBounds impl_computeGeomBounds(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)`
- `protected NGNode impl_createPeer()`
- `public List impl_getAllParentStylesheets()`
- `protected void impl_pickNodeLocal(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- `protected void impl_processCSS(javafx.beans.value.WritableValue arg0)`
- `public Object impl_processMXNode(com.sun.javafx.jmx.MXNodeAlgorithm arg0, com.sun.javafx.jmx.MXNodeAlgorithmContext arg1)`
- `public final ObjectProperty impl_traversalEngineProperty()`
- `public void impl_updatePeer()`

- `public final boolean isNeedsLayout()`
- `public final void layout()`
- `protected void layoutChildren()`
- `public Node lookup(java.lang.String arg0)`
- `public double minHeight(double arg0)`
- `public double minWidth(double arg0)`
- `public final ReadOnlyBooleanProperty needsLayoutProperty()`
- `public double prefHeight(double arg0)`
- `public double prefWidth(double arg0)`
- `public Object queryAccessibleAttribute(AccessibleAttribute arg0, java.lang.Object[] arg1)`
- `public void requestLayout()`
- `protected final void requestParentLayout()`
- `public final void setImpl_traversalEngine(com.sun.javafx.scene.traversal.ParentTraversalEngine arg0)`
- `protected final void setNeedsLayout(boolean arg0)`
- `protected void updateBounds()`

3.1.12 Members inherited from class Node

`javafx.scene.Node`

- `public final ObjectProperty accessibleHelpProperty()`
- `public final ObjectProperty accessibleRoleDescriptionProperty()`
- `public final ObjectProperty accessibleRoleProperty()`
- `public final ObjectProperty accessibleTextProperty()`
- `public final void addEventFilter(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `public final void addEventHandler(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `public final void applyCss()`
- `public final void autosize()`
- `public static final BASELINE_OFFSET_SAME_AS_HEIGHT`
- `public final ObjectProperty blendModeProperty()`
- `public final ReadOnlyObjectProperty boundsInLocalProperty()`
- `public final ReadOnlyObjectProperty boundsInParentProperty()`
- `public EventDispatchChain buildEventDispatchChain(javafx.event.EventDispatchChain arg0)`
- `public final ObjectProperty cacheHintProperty()`
- `public final BooleanProperty cacheProperty()`
- `public final ObjectProperty clipProperty()`
- `public double computeAreaInScreen()`
- `public boolean contains(double arg0, double arg1)`
- `public boolean contains(javafx.geometry.Point2D arg0)`
- `protected boolean containsBounds(double arg0, double arg1)`
- `public final ObjectProperty cursorProperty()`
- `public final ObjectProperty depthTestProperty()`
- `public final ReadOnlyBooleanProperty disabledProperty()`
- `public final BooleanProperty disableProperty()`
- `public final ReadOnlyObjectProperty effectiveNodeOrientationProperty()`
- `public final ObjectProperty effectProperty()`
- `public final ObjectProperty eventDispatcherProperty()`

- public void executeAccessibleAction(AccessibleAction arg0,
java.lang.Object[] arg1)
- public final void fireEvent(javafx.event.Event arg0)
- public final ReadOnlyBooleanProperty focusedProperty()
- public final BooleanProperty focusTraversableProperty()
- public final String getAccessibleHelp()
- public final AccessibleRole getAccessibleRole()
- public final String getAccessibleRoleDescription()
- public final String getAccessibleText()
- public double getBaselineOffset()
- public final BlendMode getBlendMode()
- public final Bounds getBoundsInLocal()
- public final Bounds getBoundsInParent()
- public final CacheHint getCacheHint()
- public static List getClassCssMetaData()
- public final Node getClip()
- public Orientation getContentBias()
- public List.getCssMetaData()
- public final Cursor getCursor()
- public final DepthTest getDepthTest()
- public final Effect getEffect()
- public final NodeOrientation getEffectiveNodeOrientation()
- public final EventDispatcher getEventDispatcher()
- public final String getId()
- public final InputMethodRequests getInputMethodRequests()
- public final Bounds getLayoutBounds()
- public final double getLayoutX()
- public final double getLayoutY()
- public final Transform getLocalToParentTransform()
- public final Transform getLocalToSceneTransform()
- public final NodeOrientation getNodeOrientation()
- public final EventHandler getOnContextMenuRequested()
- public final EventHandler getOnDragDetected()
- public final EventHandler getOnDragDone()
- public final EventHandler getOnDragDropped()
- public final EventHandler getOnDragEntered()
- public final EventHandler getOnDragExited()
- public final EventHandler getOnDragOver()
- public final EventHandler getOnInputMethodTextChanged()
- public final EventHandler getOnKeyPressed()
- public final EventHandler getOnKeyReleased()
- public final EventHandler getOnKeyTyped()
- public final EventHandler getOnMouseClicked()
- public final EventHandler getOnMouseDragEntered()
- public final EventHandler getOnMouseDragExited()
- public final EventHandler getOnMouseDragged()
- public final EventHandler getOnMouseDragOver()
- public final EventHandler getOnMouseDragReleased()
- public final EventHandler getOnMouseEntered()
- public final EventHandler getOnMouseExited()
- public final EventHandler getOnMouseMoved()
- public final EventHandler getOnMousePressed()
- public final EventHandler getOnMouseReleased()

- public final EventHandler `getOnRotate()`
- public final EventHandler `getOnRotationFinished()`
- public final EventHandler `getOnRotationStarted()`
- public final EventHandler `getOnScroll()`
- public final EventHandler `getOnScrollFinished()`
- public final EventHandler `getOnScrollStarted()`
- public final EventHandler `getOnSwipeDown()`
- public final EventHandler `getOnSwipeLeft()`
- public final EventHandler `getOnSwipeRight()`
- public final EventHandler `getOnSwipeUp()`
- public final EventHandler `getOnTouchMoved()`
- public final EventHandler `getOnTouchPressed()`
- public final EventHandler `getOnTouchReleased()`
- public final EventHandler `getOnTouchStationary()`
- public final EventHandler `getOnZoom()`
- public final EventHandler `getOnZoomFinished()`
- public final EventHandler `getOnZoomStarted()`
- public final double `getOpacity()`
- public final Parent `getParent()`
- public final ObservableMap `getProperties()`
- public final ObservableSet `getPseudoClassStates()`
- public final double `getRotate()`
- public final Point3D `getRotationAxis()`
- public final double `getScaleX()`
- public final double `getScaleY()`
- public final double `getScaleZ()`
- public final Scene `getScene()`
- public final String `getStyle()`
- public Styleable `getStyleableParent()`
- public final ObservableList `getStyleClass()`
- public final ObservableList `getTransforms()`
- public final double `getTranslateX()`
- public final double `getTranslateY()`
- public final double `getTranslateZ()`
- public String `getTypeSelector()`
- public Object `getUserData()`
- public boolean `hasProperties()`
- public final ReadOnlyBooleanProperty `hoverProperty()`
- public final StringProperty `idProperty()`
- protected final void `impl_clearDirty(com.sun.javafx.scene.DirtyBits arg0)`
- protected abstract boolean `impl_computeContains(double arg0, double arg1)`
- public abstract BaseBounds `impl_computeGeomBounds(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)`
- protected boolean `impl_computeIntersects(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- protected Bounds `impl_computeLayoutBounds()`
- protected abstract NGNode `impl_createPeer()`
- protected Cursor `impl_cssGetCursorInitialValue()`
- protected Boolean `impl_cssGetFocusTraversableInitialValue()`
- public Map `impl_findStyles(java.util.Map arg0)`

- protected void **impl_geomChanged()**
- public final BaseTransform **impl_getLeafTransform()**
- public static List **impl_getMatchingStyles**(javafx.css.CssMetaData arg0, javafx.css.Styleable arg1)
- public NGNode **impl_getPeer()**
- public final double **impl_getPivotX()**
- public final double **impl_getPivotY()**
- public final double **impl_getPivotZ()**
- public final ObservableMap **impl_getStyleMap()**
- public boolean **impl_hasTransforms()**
- protected final boolean **impl_intersects**(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)
- protected final double **impl_intersectsBounds**(com.sun.javafx.geom.PickRay arg0)
- protected final boolean **impl_isDirty**(com.sun.javafx.scene.DirtyBits arg0)
- protected final boolean **impl_isDirtyEmpty()**
- public final boolean **impl_isShowMnemonics()**
- public final boolean **impl_isTreeVisible()**
- protected final void **impl_layoutBoundsChanged()**
- protected void **impl_markDirty**(com.sun.javafx.scene.DirtyBits arg0)
- protected void **impl_notifyLayoutBoundsChanged()**
- public final void **impl_pickNode**(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)
- protected void **impl_pickNodeLocal**(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)
- public final void **impl_processCSS**(boolean arg0)
- protected void **impl_processCSS**(javafx.beans.value.WritableValue arg0)
- public abstract Object **impl_processMXNode**(com.sun.javafx.jmx.MXNodeAlgorithm arg0, com.sun.javafx.jmx.MXNodeAlgorithmContext arg1)
- public final void **impl_reapplyCSS()**
- public final void **impl_setShowMnemonics**(boolean arg0)
- public final void **impl_setStyleMap**(javafx.collections.ObservableMap arg0)
- public final BooleanProperty **impl_showMnemonicsProperty()**
- public final void **impl_syncPeer()**
- public void **impl_transformsChanged()**
- public final boolean **impl_traverse**(com.sun.javafx.scene.traversal.Direction arg0)
- protected final BooleanExpression **impl_treeVisibleProperty()**
- public void **impl_updatePeer()**
- public final ObjectProperty **inputMethodRequestsProperty()**
- public boolean **intersects**(javafx.geometry.Bounds arg0)
- public boolean **intersects**(double arg0, double arg1, double arg2, double arg3)
- public final boolean **isCache()**
- public final boolean **isDisable()**
- public final boolean **isDisabled()**
- public final boolean **isFocused()**
- public final boolean **isFocusTraversable()**

- public final boolean isHover()
- public final boolean isManaged()
- public final boolean isMouseTransparent()
- public final boolean isPickOnBounds()
- public final boolean isPressed()
- public boolean isResizable()
- public final boolean isVisible()
- public final ReadOnlyObjectProperty layoutBoundsProperty()
- public final DoubleProperty layoutXProperty()
- public final DoubleProperty layoutYProperty()
- public Bounds localToParent(javafx.geometry.Bounds arg0)
- public Point2D localToParent(double arg0, double arg1)
- public Point3D localToParent(double arg0, double arg1, double arg2)
- public Point2D localToParent(javafx.geometry.Point2D arg0)
- public Point3D localToParent(javafx.geometry.Point3D arg0)
- public final ReadOnlyObjectProperty localToParentTransformProperty()
- public Bounds localToScene(javafx.geometry.Bounds arg0)
- public Bounds localToScene(javafx.geometry.Bounds arg0, boolean arg1)
- public Point2D localToScene(double arg0, double arg1)
- public Point2D localToScene(double arg0, double arg1, boolean arg2)
- public Point3D localToScene(double arg0, double arg1, double arg2)
- public Point3D localToScene(double arg0, double arg1, double arg2, boolean arg3)
- public Point2D localToScene(javafx.geometry.Point2D arg0)
- public Point2D localToScene(javafx.geometry.Point2D arg0, boolean arg1)
- public Point3D localToScene(javafx.geometry.Point3D arg0)
- public Point3D localToScene(javafx.geometry.Point3D arg0, boolean arg1)
- public final ReadOnlyObjectProperty localToSceneTransformProperty()
- public Bounds localToScreen(javafx.geometry.Bounds arg0)
- public Point2D localToScreen(double arg0, double arg1)
- public Point2D localToScreen(double arg0, double arg1, double arg2)
- public Point2D localToScreen(javafx.geometry.Point2D arg0)
- public Point2D localToScreen(javafx.geometry.Point3D arg0)
- public Node lookup(java.lang.String arg0)
- public Set lookupAll(java.lang.String arg0)
- public final BooleanProperty managedProperty()
- public double maxHeight(double arg0)
- public double maxWidth(double arg0)
- public double minHeight(double arg0)
- public double minWidth(double arg0)
- public final BooleanProperty mouseTransparentProperty()
- public final ObjectProperty nodeOrientationProperty()

- `public final void notifyAccessibleAttributeChanged(AccessibleAttribute arg0)`
- `public final ObjectProperty onContextMenuRequestedProperty()`
- `public final ObjectProperty onDragDetectedProperty()`
- `public final ObjectProperty onDragDoneProperty()`
- `public final ObjectProperty onDragDroppedProperty()`
- `public final ObjectProperty onDragEnteredProperty()`
- `public final ObjectProperty onDragExitedProperty()`
- `public final ObjectProperty onDragOverProperty()`
- `public final ObjectProperty onInputMethodTextChangedProperty()`
- `public final ObjectProperty onKeyPressedProperty()`
- `public final ObjectProperty onKeyReleasedProperty()`
- `public final ObjectProperty onKeyTypedProperty()`
- `public final ObjectProperty onMouseClickedProperty()`
- `public final ObjectProperty onMouseDragEnteredProperty()`
- `public final ObjectProperty onMouseDragExitedProperty()`
- `public final ObjectProperty onMouseDraggedProperty()`
- `public final ObjectProperty onMouseDragOverProperty()`
- `public final ObjectProperty onMouseDragReleasedProperty()`
- `public final ObjectProperty onMouseEnteredProperty()`
- `public final ObjectProperty onMouseExitedProperty()`
- `public final ObjectProperty onMouseMovedProperty()`
- `public final ObjectProperty onMousePressedProperty()`
- `public final ObjectProperty onMouseReleasedProperty()`
- `public final ObjectProperty onRotateProperty()`
- `public final ObjectProperty onRotationFinishedProperty()`
- `public final ObjectProperty onRotationStartedProperty()`
- `public final ObjectProperty onScrollFinishedProperty()`
- `public final ObjectProperty onScrollProperty()`
- `public final ObjectProperty onScrollStartedProperty()`
- `public final ObjectProperty onSwipeDownProperty()`
- `public final ObjectProperty onSwipeLeftProperty()`
- `public final ObjectProperty onSwipeRightProperty()`
- `public final ObjectProperty onSwipeUpProperty()`
- `public final ObjectProperty onTouchMovedProperty()`
- `public final ObjectProperty onTouchPressedProperty()`
- `public final ObjectProperty onTouchReleasedProperty()`
- `public final ObjectProperty onTouchStationaryProperty()`
- `public final ObjectProperty onZoomFinishedProperty()`
- `public final ObjectProperty onZoomProperty()`
- `public final ObjectProperty onZoomStartedProperty()`
- `public final DoubleProperty opacityProperty()`
- `public final ReadOnlyObjectProperty parentProperty()`
- `public Bounds parentToLocal(javafx.geometry.Bounds arg0)`
- `public Point2D parentToLocal(double arg0, double arg1)`
- `public Point3D parentToLocal(double arg0, double arg1, double arg2)`
- `public Point2D parentToLocal(javafx.geometry.Point2D arg0)`
- `public Point3D parentToLocal(javafx.geometry.Point3D arg0)`
- `public final BooleanProperty pickOnBoundsProperty()`

- public double **prefHeight**(double arg0)
- public double **prefWidth**(double arg0)
- public final ReadOnlyBooleanProperty **pressedProperty**()
- public final void **pseudoClassStateChanged**(javafx.css.PseudoClass arg0, boolean arg1)
- public Object **queryAccessibleAttribute**(AccessibleAttribute arg0, java.lang.Object[] arg1)
- public void **relocate**(double arg0, double arg1)
- public final void **removeEventFilter**(javafx.event.EventType arg0, javafx.event.EventHandler arg1)
- public final void **removeEventHandler**(javafx.event.EventType arg0, javafx.event.EventHandler arg1)
- public void **requestFocus**()
- public void **resize**(double arg0, double arg1)
- public void **resizeRelocate**(double arg0, double arg1, double arg2, double arg3)
- public final DoubleProperty **rotateProperty**()
- public final ObjectProperty **rotationAxisProperty**()
- public final DoubleProperty **scaleXProperty**()
- public final DoubleProperty **scaleYProperty**()
- public final DoubleProperty **scaleZProperty**()
- public final ReadOnlyObjectProperty **sceneProperty**()
- public Bounds **sceneToLocal**(javafx.geometry.Bounds arg0)
- public Bounds **sceneToLocal**(javafx.geometry.Bounds arg0, boolean arg1)
- public Point2D **sceneToLocal**(double arg0, double arg1)
- public Point2D **sceneToLocal**(double arg0, double arg1, boolean arg2)
- public Point3D **sceneToLocal**(double arg0, double arg1, double arg2)
- public Point2D **sceneToLocal**(javafx.geometry.Point2D arg0)
- public Point2D **sceneToLocal**(javafx.geometry.Point2D arg0, boolean arg1)
- public Point3D **sceneToLocal**(javafx.geometry.Point3D arg0)
- public Bounds **screenToLocal**(javafx.geometry.Bounds arg0)
- public Point2D **screenToLocal**(double arg0, double arg1)
- public Point2D **screenToLocal**(javafx.geometry.Point2D arg0)
- public final void **setAccessibleHelp**(java.lang.String arg0)
- public final void **setAccessibleRole**(AccessibleRole arg0)
- public final void **setAccessibleRoleDescription**(java.lang.String arg0)
- public final void **setAccessibleText**(java.lang.String arg0)
- public final void **setBlendMode**(effect.BlendMode arg0)
- public final void **setCache**(boolean arg0)
- public final void **setCacheHint**(CacheHint arg0)
- public final void **setClip**(Node arg0)
- public final void **setCursor**(Cursor arg0)
- public final void **setDepthTest**(DepthTest arg0)
- public final void **setDisable**(boolean arg0)
- protected final void **setDisabled**(boolean arg0)
- public final void **setEffect**(effect.Effect arg0)

- `public final void setEventDispatcher(javafx.event.EventDispatcher arg0)`
- `protected final void setEventHandler(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `protected final void setFocused(boolean arg0)`
- `public final void setFocusTraversable(boolean arg0)`
- `protected final void setHover(boolean arg0)`
- `public final void setId(java.lang.String arg0)`
- `public final void setInputMethodRequests(input.InputMethodRequests arg0)`
- `public final void setLayoutX(double arg0)`
- `public final void setLayoutY(double arg0)`
- `public final void setManaged(boolean arg0)`
- `public final void setMouseTransparent(boolean arg0)`
- `public final void setNodeOrientation(javafx.geometry.NodeOrientation arg0)`
- `public final void setOnContextMenuRequested(javafx.event.EventHandler arg0)`
- `public final void setOnDragDetected(javafx.event.EventHandler arg0)`
- `public final void setOnDragDone(javafx.event.EventHandler arg0)`
- `public final void setOnDragDropped(javafx.event.EventHandler arg0)`
- `public final void setOnDragEntered(javafx.event.EventHandler arg0)`
- `public final void setOnDragExited(javafx.event.EventHandler arg0)`
- `public final void setOnDragOver(javafx.event.EventHandler arg0)`
- `public final void setOnInputMethodTextChanged(javafx.event.EventHandler arg0)`
- `public final void setOnKeyPressed(javafx.event.EventHandler arg0)`
- `public final void setOnKeyReleased(javafx.event.EventHandler arg0)`
- `public final void setOnKeyTyped(javafx.event.EventHandler arg0)`
- `public final void setOnMouseClicked(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragEntered(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragExited(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragged(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragOver(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragReleased(javafx.event.EventHandler arg0)`

- `public final void setOnMouseEntered(javafx.event.EventHandler arg0)`
- `public final void setOnMouseExited(javafx.event.EventHandler arg0)`
- `public final void setOnMouseMoved(javafx.event.EventHandler arg0)`
- `public final void setOnMousePressed(javafx.event.EventHandler arg0)`
- `public final void setOnMouseReleased(javafx.event.EventHandler arg0)`
- `public final void setOnRotate(javafx.event.EventHandler arg0)`
- `public final void setOnRotationFinished(javafx.event.EventHandler arg0)`
- `public final void setOnRotationStarted(javafx.event.EventHandler arg0)`
- `public final void setOnScroll(javafx.event.EventHandler arg0)`
- `public final void setOnScrollFinished(javafx.event.EventHandler arg0)`
- `public final void setOnScrollStarted(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeDown(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeLeft(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeRight(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeUp(javafx.event.EventHandler arg0)`
- `public final void setOnTouchMoved(javafx.event.EventHandler arg0)`
- `public final void setOnTouchPressed(javafx.event.EventHandler arg0)`
- `public final void setOnTouchReleased(javafx.event.EventHandler arg0)`
- `public final void setOnTouchStationary(javafx.event.EventHandler arg0)`
- `public final void setOnZoom(javafx.event.EventHandler arg0)`
- `public final void setOnZoomFinished(javafx.event.EventHandler arg0)`
- `public final void setOnZoomStarted(javafx.event.EventHandler arg0)`
- `public final void setOpacity(double arg0)`
- `public final void setPickOnBounds(boolean arg0)`
- `protected final void setPressed(boolean arg0)`
- `public final void setRotate(double arg0)`
- `public final void setRotationAxis(javafx.geometry.Point3D arg0)`
- `public final void setScaleX(double arg0)`
- `public final void setScaleY(double arg0)`
- `public final void setScaleZ(double arg0)`
- `public final void setStyle(java.lang.String arg0)`

- `public final void setTranslateX(double arg0)`
- `public final void setTranslateY(double arg0)`
- `public final void setTranslateZ(double arg0)`
- `public void setUserData(java.lang.Object arg0)`
- `public final void setVisible(boolean arg0)`
- `public void snapshot(javafx.util.Callback arg0, SnapshotParameters arg1, image.WritableImage arg2)`
- `public WritableImage snapshot(SnapshotParameters arg0, image.WritableImage arg1)`
- `public Dragboard startDragAndDrop(input.TransferMode[] arg0)`
- `public void startFullDrag()`
- `public final StringProperty styleProperty()`
- `public void toBack()`
- `public void toFront()`
- `public String toString()`
- `public final DoubleProperty translateXProperty()`
- `public final DoubleProperty translateYProperty()`
- `public final DoubleProperty translateZProperty()`
- `public boolean usesMirroring()`
- `public final BooleanProperty visibleProperty()`

3.2 Class DetailFenetre

Gre l’affichage sous forme textuelle des details d’une fenetre de livraison dans la `TreeTableView`.

3.2.1 Declaration

```
public class DetailFenetre
    extends vue.ObjetVisualisable
```

3.2.2 Constructor summary

[DetailFenetre\(Fenetre\)](#) Constructeur du dtail fenetre

3.2.3 Method summary

[afficherCaracteriquesGlobales\(\)](#)
[afficherCaracteriquesSpeciales\(\)](#)
[getFenetre\(\)](#)

3.2.4 Constructors

- **DetailFenetre**

```
public DetailFenetre (modele.donneesxml.Fenetre
    fenetre)
```

- **Description**

Constructeur du dtail fentre

- **Parameters**

* **fenetre** – La fentre associe

3.2.5 Methods

- **afficherCaracteriquesGlobales**

```
public abstract java.lang.String
    afficherCaracteriquesGlobales ()
```

- **Returns** – Les caractristiques globales de l'lment affich

- **afficherCaracteriquesSpeciales**

```
public abstract java.lang.String
    afficherCaracteriquesSpeciales ()
```

- **Returns** – Les caractristiques spciales de l'lment affich

- **getFenetre**

```
public modele.donneesxml.Fenetre getFenetre ()
```

- **Returns** – La fentre associe ce dtail

3.2.6 Members inherited from class **ObjetVisualisable**

vue.ObjetVisualisable

- public abstract String **afficherCaracteriquesGlobales()**
- public abstract String **afficherCaracteriquesSpeciales()**
- protected static String **convertirEnHeureLisible(int tempsEnSeconde)**
- public **ObjetVisualisable.CouleurTexte** **getCouleurDefaut()**
- public void **setCouleurDefaut(ObjetVisualisable.CouleurTexte couleur)**

3.3 Class DetailLivraison

Gre l’affichage sous forme textuelle des details d’une livraison dans la TreeTableView.

3.3.1 Declaration

```
public class DetailLivraison
    extends vue.ObjetVisualisable
```

3.3.2 Constructor summary

DetailLivraison(Livraison) Constructeur du dtail de livraison

3.3.3 Method summary

afficherCaracteriquesGlobales()
afficherCaracteriquesSpeciales()
getLivraison()

3.3.4 Constructors

- **DetailLivraison**

```
public DetailLivraison (modele.donneesxml.Livraison
    livraison)
```

- **Description**

Constructeur du dtail de livraison

- **Parameters**

* livraison – La livraison associe

3.3.5 Methods

- **afficherCaracteriquesGlobales**

```
public abstract java.lang.String
    afficherCaracteriquesGlobales()
```

- **Returns** – Les caractristiques globales de l’lment affich

- **afficherCaracteriquesSpeciales**

```
public abstract java.lang.String
    afficherCaracteriquesSpeciales()
```

– **Returns** – Les caractéristiques spéciales de l'élément affiché

- **getLivraison**

```
public modele.donneesxml.Livraison getLivraison()
```

– **Returns** – La livraison associée

3.3.6 Members inherited from class `ObjetVisualisable`

`vue.ObjetVisualisable`

- `public abstract String afficherCaracteriquesGlobales()`
- `public abstract String afficherCaracteriquesSpeciales()`
- `protected static String convertirEnHeureLisible(int tempsEnSeconde)`
- `public ObjetVisualisable.CouleurTexte getCouleurDefaut()`
- `public void setCouleurDefaut(ObjetVisualisable.CouleurTexte couleur)`

3.4 Class `FenetrePrincipale`

Cette classe crée la fenêtre principale avec ses enfants. Elle se charge aussi de créer le contrôleur.

3.4.1 Declaration

```
public class FenetrePrincipale
    extends javafx.application.Application
```

3.4.2 Constructor summary

[`FenetrePrincipale\(\)`](#)

3.4.3 Method summary

[`start\(Stage\)`](#) Lance l'application

3.4.4 Constructors

- **FenetrePrincipale**

```
public FenetrePrincipale()
```

3.4.5 Methods

- **start**

```
public void start(javaafx.stage.Stage primaryStage)
    throws java.lang.Exception
```

- **Description**

Lance l'application

- **Parameters**

* **primaryStage** – objet pre qui contient la fenetre principale

3.4.6 Members inherited from class Application

javafx.application.Application

- **public final** HostServices **getHostServices()**
- **public final** Application.Parameters **getParameters()**
- **public static** String **getUserAgentStylesheet()**
- **public void** **init()** **throws** java.lang.Exception
- **public static void** **launch**(java.lang.Class **arg0**, java.lang.String[] **arg1**)
- **public static void** **launch**(java.lang.String[] **arg0**)
- **public final void** **notifyPreloader**(Preloader.PreloaderNotification **arg0**)
- **public static void** **setUserAgentStylesheet**(java.lang.String **arg0**)
- **public abstract void** **start**(javafx.stage.Stage **arg0**) **throws** java.lang.Exception
- **public void** **stop()** **throws** java.lang.Exception
- **public static final** **STYLESHEET_CASPIAN**
- **public static final** **STYLESHEET_MODENA**

3.5 Class ObjetVisualisable

Cette classe permet de visualiser une fentre (de livraison) ou une livraison sous forme textuelle. Cette visualisation peut se faire de deux faons faons diffrentes en fonction du type concret de l'objet.

3.5.1 Declaration

```
public abstract class ObjetVisualisable
    extends java.lang.Object
```

3.5.2 All known subclasses

DetailFenetre , DetailLivraison

3.5.3 Constructor summary

[Objet Visualisable\(\)](#)

3.5.4 Method summary

[afficherCaracteriquesGlobales\(\)](#)

[afficherCaracteriquesSpeciales\(\)](#)

[convertirEnHeureLisible\(int\)](#) Convertit un temps en seconde
en chaine de caractre sous la forme HH:mm:ss

[getCouleurDefault\(\)](#)

[setCouleurDefault\(Objet Visualisable.CouleurTexte\)](#)

3.5.5 Constructors

- **Objet Visualisable**

```
public  ObjetVisualisable ()
```

3.5.6 Methods

- **afficherCaracteriquesGlobales**

```
public abstract java.lang.String  
    afficherCaracteriquesGlobales ()
```

– **Returns** – Les caractristiques globales de l’lment affich

- **afficherCaracteriquesSpeciales**

```
public abstract java.lang.String  
    afficherCaracteriquesSpeciales ()
```

– **Returns** – Les caractristiques spciales de l’lment affich

- **convertirEnHeureLisible**

```
protected static java.lang.String  
    convertirEnHeureLisible (int tempsEnSeconde)
```

– **Description**

Convertit un temps en seconde en chaine de caractre sous la forme
HH:mm:ss

– **Parameters**

* `tempsEnSeconde` – temps à convertir

- **`getCouleurDefault`**

```
public ObjetVisualisable.CouleurTexte  
    getCouleurDefault()
```

– **Returns** – Retourne la couleur actuelle

- **`setCouleurDefault`**

```
public void setCouleurDefault(ObjetVisualisable.  
    CouleurTexte couleur)
```

– **Parameters**

* `couleur` – La nouvelle couleur

3.6 Class `ObjetVisualisable.CouleurTexte`

Différentes couleurs possibles pour un élément dans la liste.

3.6.1 Declaration

```
public static final class ObjetVisualisable.  
    CouleurTexte  
    extends java.lang.Enum
```

3.6.2 Field summary

[`NON_SURBRILLANCE`](#)
[`RETARD`](#)
[`SURBRILLANCE`](#)

3.6.3 Method summary

[`valueOf\(String\)`](#)
[`values\(\)`](#)

3.6.4 Fields

- `public static final ObjetVisualisable.CouleurTexte SURBRILLANCE`
- `public static final ObjetVisualisable.CouleurTexte NON_SURBRILLANCE`
- `public static final ObjetVisualisable.CouleurTexte RETARD`

3.6.5 Methods

- `valueOf`

```
public static ObjetVisualisable.CouleurTexte  
    valueOf(java.lang.String name)
```

- `values`

```
public static ObjetVisualisable.CouleurTexte []  
    values()
```

3.6.6 Members inherited from class Enum

`java.lang.Enum`

- `protected final Object clone() throws CloneNotSupportedException`
- `public final int compareTo(Enum arg0)`
- `public final boolean equals(Object arg0)`
- `protected final void finalize()`
- `public final Class getDeclaringClass()`
- `public final int hashCode()`
- `public final String name()`
- `public final int ordinal()`
- `public String toString()`
- `public static Enum valueOf(Class arg0, String arg1)`

3.7 Class ObservateurMessageChamps

Champ texte, couteur des messages qui peuvent tre reu

3.7.1 Declaration

```
public class ObservateurMessageChamps
    extends javafx.scene.text.Text implements controleur.
        observateur.MessageObservateur
```

3.7.2 Constructor summary

[ObservateurMessageChamps\(\)](#)

3.7.3 Method summary

[notifierObservateursMessage\(String\)](#)

3.7.4 Constructors

- ObservateurMessageChamps

```
public ObservateurMessageChamps()
```

3.7.5 Methods

- notifierObservateursMessage

```
void notifierObservateursMessage(java.lang.String
    message)
```

- **Description** copied from [controleur.observateur.MessageObservateur](#)

Notifie les observateurs qu'il doit afficher un nouveau message

- **Parameters**

* message – Le message envoy

3.7.6 Members inherited from class Text

```
javafx.scene.text.Text
```

- public final ReadOnlyDoubleProperty **baselineOffsetProperty()**
- public final ObjectProperty **boundsTypeProperty()**
- public final ObjectProperty **fontProperty()**
- public final ObjectProperty **fontSmoothingTypeProperty()**
- public final double **getBaselineOffset()**
- public final TextBoundsType **getBoundsType()**
- public static List **getClassCssMetaData()**
- public List **getCssMetaData()**
- public final Font **getFont()**

- public final FontSmoothingType getFontSmoothingType()
- public final int getImpl_caretPosition()
- public final PathElement getImpl_caretShape()
- public final int getImpl_selectionEnd()
- public final PathElement getImpl_selectionShape()
- public final int getImpl_selectionStart()
- public final double getLineSpacing()
- public final String getText()
- public final TextAlignment getTextAlignment()
- public final VPos getTextOrigin()
- public final double getWrappingWidth()
- public final double getX()
- public final double getY()
- public final BooleanProperty impl_caretBiasProperty()
- public final IntegerProperty impl_caretPositionProperty()
- public final ReadOnlyObjectProperty impl_caretShapeProperty()
- protected final boolean impl_computeContains(double arg0, double arg1)
- public final BaseBounds impl_computeGeomBounds(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)
- protected final Bounds impl_computeLayoutBounds()
- public final Shape impl_configShape()
- protected final NGNode impl_createPeer()
- public final void impl_displaySoftwareKeyboard(boolean arg0)
- protected final void impl_geomChanged()
- public final PathElement impl_getRangeShape(int arg0, int arg1)
- public final PathElement impl_getUnderlineShape(int arg0, int arg1)
- public final HitInfo impl_hitTestChar(javafx.geometry.Point2D arg0)
- public final IntegerProperty impl_selectionEndProperty()
- public final ObjectProperty impl_selectionFillProperty()
- public final ReadOnlyObjectProperty impl_selectionShapeProperty()
- public final IntegerProperty impl_selectionStartProperty()
- public final void impl_updatePeer()
- public final boolean isImpl_caretBias()
- public final boolean isStrikethrough()
- public final boolean isUnderline()
- public final DoubleProperty lineSpacingProperty()
- public Object queryAccessibleAttribute(javafx.scene.AccessibleAttribute arg0, java.lang.Object[] arg1)
- public final void setBoundsType(TextBoundsType arg0)
- public final void setFont(Font arg0)
- public final void setFontSmoothingType(FontSmoothingType arg0)
- public final void setImpl_caretBias(boolean arg0)
- public final void setImpl_caretPosition(int arg0)
- public final void setImpl_selectionEnd(int arg0)
- public final void setImpl_selectionStart(int arg0)
- public final void setLineSpacing(double arg0)
- public final void setStrikethrough(boolean arg0)

- public final void **setText**(java.lang.String arg0)
- public final void **setTextAlignment**(TextAlignment arg0)
- public final void **setTextOrigin**(javafx.geometry.VPos arg0)
- public final void **setUnderline**(boolean arg0)
- public final void **setWrappingWidth**(double arg0)
- public final void **setX**(double arg0)
- public final void **setY**(double arg0)
- public final BooleanProperty **strikethroughProperty**()
- public final ObjectProperty **textAlignmentProperty**()
- public final ObjectProperty **textOriginProperty**()
- public final StringProperty **textProperty**()
- public String **toString**()
- public final BooleanProperty **underlineProperty**()
- public boolean **usesMirroring**()
- public final DoubleProperty **wrappingWidthProperty**()
- public final DoubleProperty **xProperty**()
- public final DoubleProperty **yProperty**()

3.7.7 Members inherited from class Shape

`javafx.scene.shape.Shape`

- public final ObjectProperty **fillProperty**()
- public static List **getClassCssMetaData**()
- public List **getCssMetaData**()
- public final Paint **getFill**()
- public final Paint **getStroke**()
- public final ObservableList **getStrokeDashArray**()
- public final double **getStrokeDashOffset**()
- public final StrokeLineCap **getStrokeLineCap**()
- public final StrokeLineJoin **getStrokeLineJoin**()
- public final double **getStrokeMiterLimit**()
- public final StrokeType **getStrokeType**()
- public final double **getStrokeWidth**()
- protected boolean **impl_computeContains**(double arg0, double arg1)
- public BaseBounds **impl_computeGeomBounds**(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)
- public abstract Shape **impl_configShape**()
- protected NGNode **impl_createPeer**()
- protected Paint **impl_cssGetFillInitialValue**()
- protected Paint **impl_cssGetStrokeInitialValue**()
- protected void **impl_markDirty**(com.sun.javafx.scene.DirtyBits arg0)
- protected **impl_mode**
- public Object **impl_processMXNode**(com.sun.javafx.jmx.MXNodeAlgorithm arg0, com.sun.javafx.jmx.MXNodeAlgorithmContext arg1)
- public void **impl_setShapeChangeListener**(java.lang.Runnable arg0)
- public void **impl_updatePeer**()
- public static Shape **intersect**(Shape arg0, Shape arg1)
- public final boolean **isSmooth**()
- public final void **setFill**(javafx.scene.paint.Paint arg0)

- `public final void setSmooth(boolean arg0)`
- `public final void setStroke(javafx.scene.paint.Paint arg0)`
- `public final void setStrokeDashOffset(double arg0)`
- `public final void setStrokeLineCap(StrokeLineCap arg0)`
- `public final void setStrokeLineJoin(StrokeLineJoin arg0)`
- `public final void setStrokeMiterLimit(double arg0)`
- `public final void setStrokeType(StrokeType arg0)`
- `public final void setStrokeWidth(double arg0)`
- `public final BooleanProperty smoothProperty()`
- `public final DoubleProperty strokeDashOffsetProperty()`
- `public final ObjectProperty strokeLineCapProperty()`
- `public final ObjectProperty strokeLineJoinProperty()`
- `public final DoubleProperty strokeMiterLimitProperty()`
- `public final ObjectProperty strokeProperty()`
- `public final ObjectProperty strokeTypeProperty()`
- `public final DoubleProperty strokeWidthProperty()`
- `public static Shape subtract(Shape arg0, Shape arg1)`
- `public static Shape union(Shape arg0, Shape arg1)`

3.7.8 Members inherited from class Node

`javafx.scene.Node`

- `public final ObjectProperty accessibleHelpProperty()`
- `public final ObjectProperty accessibleRoleDescriptionProperty()`
- `public final ObjectProperty accessibleRoleProperty()`
- `public final ObjectProperty accessibleTextProperty()`
- `public final void addEventFilter(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `public final void addEventHandler(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `public final void applyCss()`
- `public final void autosize()`
- `public static final BASELINE_OFFSET_SAME_AS_HEIGHT`
- `public final ObjectProperty blendModeProperty()`
- `public final ReadOnlyObjectProperty boundsInLocalProperty()`
- `public final ReadOnlyObjectProperty boundsInParentProperty()`
- `public EventDispatchChain buildEventDispatchChain(javafx.event.EventDispatchChain arg0)`
- `public final ObjectProperty cacheHintProperty()`
- `public final BooleanProperty cacheProperty()`
- `public final ObjectProperty clipProperty()`
- `public double computeAreaInScreen()`
- `public boolean contains(double arg0, double arg1)`
- `public boolean contains(javafx.geometry.Point2D arg0)`
- `protected boolean containsBounds(double arg0, double arg1)`
- `public final ObjectProperty cursorProperty()`
- `public final ObjectProperty depthTestProperty()`
- `public final ReadOnlyBooleanProperty disabledProperty()`
- `public final BooleanProperty disableProperty()`
- `public final ReadOnlyObjectProperty effectiveNodeOrientationProperty()`
- `public final ObjectProperty effectProperty()`

- public final ObjectProperty eventDispatcherProperty()
- public void executeAccessibleAction(AccessibleAction arg0,
java.lang.Object[] arg1)
- public final void fireEvent(javafx.event.Event arg0)
- public final ReadOnlyBooleanProperty focusedProperty()
- public final BooleanProperty focusTraversableProperty()
- public final String getAccessibleHelp()
- public final AccessibleRole getAccessibleRole()
- public final String getAccessibleRoleDescription()
- public final String getAccessibleText()
- public double getBaselineOffset()
- public final BlendMode getBlendMode()
- public final Bounds getBoundsInLocal()
- public final Bounds getBoundsInParent()
- public final CacheHint getCacheHint()
- public static List getClassCssMetaData()
- public final Node getClip()
- public Orientation getContentBias()
- public List getCssMetaData()
- public final Cursor getCursor()
- public final DepthTest getDepthTest()
- public final Effect getEffect()
- public final NodeOrientation getEffectiveNodeOrientation()
- public final EventDispatcher getEventDispatcher()
- public final String getId()
- public final InputMethodRequests getInputMethodRequests()
- public final Bounds getLayoutBounds()
- public final double getLayoutX()
- public final double getLayoutY()
- public final Transform getLocalToParentTransform()
- public final Transform getLocalToSceneTransform()
- public final NodeOrientation getNodeOrientation()
- public final EventHandler getOnContextMenuRequested()
- public final EventHandler getOnDragDetected()
- public final EventHandler getOnDragDone()
- public final EventHandler getOnDragDropped()
- public final EventHandler getOnDragEntered()
- public final EventHandler getOnDragExited()
- public final EventHandler getOnDragOver()
- public final EventHandler getOnInputMethodTextChanged()
- public final EventHandler getOnKeyPressed()
- public final EventHandler getOnKeyReleased()
- public final EventHandler getOnKeyTyped()
- public final EventHandler getOnMouseClicked()
- public final EventHandler getOnMouseDragEntered()
- public final EventHandler getOnMouseDragExited()
- public final EventHandler getOnMouseDragged()
- public final EventHandler getOnMouseDragOver()
- public final EventHandler getOnMouseDragReleased()
- public final EventHandler getOnMouseEntered()
- public final EventHandler getOnMouseExited()
- public final EventHandler getOnMouseMoved()
- public final EventHandler getOnMousePressed()

- public final EventHandler getOnMouseReleased()
- public final EventHandler getOnRotate()
- public final EventHandler getOnRotationFinished()
- public final EventHandler getOnRotationStarted()
- public final EventHandler getOnScroll()
- public final EventHandler getOnScrollFinished()
- public final EventHandler getOnScrollStarted()
- public final EventHandler getOnSwipeDown()
- public final EventHandler getOnSwipeLeft()
- public final EventHandler getOnSwipeRight()
- public final EventHandler getOnSwipeUp()
- public final EventHandler getOnTouchMoved()
- public final EventHandler getOnTouchPressed()
- public final EventHandler getOnTouchReleased()
- public final EventHandler getOnTouchStationary()
- public final EventHandler getOnZoom()
- public final EventHandler getOnZoomFinished()
- public final EventHandler getOnZoomStarted()
- public final double getOpacity()
- public final Parent getParent()
- public final ObservableMap getProperties()
- public final ObservableSet getPseudoClassStates()
- public final double getRotate()
- public final Point3D getRotationAxis()
- public final double getScaleX()
- public final double getScaleY()
- public final double getScaleZ()
- public final Scene getScene()
- public final String getStyle()
- public Styleable getStyleableParent()
- public final ObservableList getStyleClass()
- public final ObservableList getTransforms()
- public final double getTranslateX()
- public final double getTranslateY()
- public final double getTranslateZ()
- public String getTypeSelector()
- public Object getUserData()
- public boolean hasProperties()
- public final ReadOnlyBooleanProperty hoverProperty()
- public final StringProperty idProperty()
- protected final void impl_clearDirty(com.sun.javafx.scene.DirtyBits arg0)
- protected abstract boolean impl_computeContains(double arg0, double arg1)
- public abstract BaseBounds impl_computeGeomBounds(com.sun.javafx.geom.BaseBounds arg0, com.sun.javafx.geom.transform.BaseTransform arg1)
- protected boolean impl_computeIntersects(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)
- protected Bounds impl_computeLayoutBounds()
- protected abstract NGNode impl_createPeer()
- protected Cursor impl_cssGetCursorInitialValue()
- protected Boolean impl_cssGetFocusTraversableInitialValue()

- `public Map impl_findStyles(java.util.Map arg0)`
- `protected void impl_geomChanged()`
- `public final BaseTransform impl_getLeafTransform()`
- `public static List impl_getMatchingStyles(javafx.css.CssMetaData arg0, javafx.css.Styleable arg1)`
- `public NGNode impl_getPeer()`
- `public final double impl_getPivotX()`
- `public final double impl_getPivotY()`
- `public final double impl_getPivotZ()`
- `public final ObservableMap impl_getStyleMap()`
- `public boolean impl_hasTransforms()`
- `protected final boolean impl_intersects(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- `protected final double impl_intersectsBounds(com.sun.javafx.geom.PickRay arg0)`
- `protected final boolean impl_isDirty(com.sun.javafx.scene.DirtyBits arg0)`
- `protected final boolean impl_isDirtyEmpty()`
- `public final boolean impl_isShowMnemonics()`
- `public final boolean impl_isTreeVisible()`
- `protected final void impl_layoutBoundsChanged()`
- `protected void impl_markDirty(com.sun.javafx.scene.DirtyBits arg0)`
- `protected void impl_notifyLayoutBoundsChanged()`
- `public final void impl_pickNode(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- `protected void impl_pickNodeLocal(com.sun.javafx.geom.PickRay arg0, com.sun.javafx.scene.input.PickResultChooser arg1)`
- `public final void impl_processCSS(boolean arg0)`
- `protected void impl_processCSS(javafx.beans.value.WritableValue arg0)`
- `public abstract Object impl_processMXNode(com.sun.javafx.jmx.MXNodeAlgorithm arg0, com.sun.javafx.jmx.MXNodeAlgorithmContext arg1)`
- `public final void impl_reapplyCSS()`
- `public final void impl_setShowMnemonics(boolean arg0)`
- `public final void impl_setStyleMap(javafx.collections.ObservableMap arg0)`
- `public final BooleanProperty impl_showMnemonicsProperty()`
- `public final void impl_syncPeer()`
- `public void impl_transformsChanged()`
- `public final boolean impl_traverse(com.sun.javafx.scene.traversal.Direction arg0)`
- `protected final BooleanExpression impl_treeVisibleProperty()`
- `public void impl_updatePeer()`
- `public final ObjectProperty inputMethodRequestsProperty()`
- `public boolean intersects(javafx.geometry.Bounds arg0)`
- `public boolean intersects(double arg0, double arg1, double arg2, double arg3)`
- `public final boolean isCache()`
- `public final boolean isDisable()`
- `public final boolean isDisabled()`
- `public final boolean isFocused()`

- public final boolean **isFocusTraversable()**
- public final boolean **isHover()**
- public final boolean **isManaged()**
- public final boolean **isMouseTransparent()**
- public final boolean **isPickOnBounds()**
- public final boolean **isPressed()**
- public boolean **isResizable()**
- public final boolean **isVisible()**
- public final ReadOnlyObjectProperty **layoutBoundsProperty()**
- public final DoubleProperty **layoutXProperty()**
- public final DoubleProperty **layoutYProperty()**
- public Bounds **localToParent**(javafx.geometry.Bounds arg0)
- public Point2D **localToParent**(double arg0, double arg1)
- public Point3D **localToParent**(double arg0, double arg1, double arg2)
- public Point2D **localToParent**(javafx.geometry.Point2D arg0)
- public Point3D **localToParent**(javafx.geometry.Point3D arg0)
- public final ReadOnlyObjectProperty **localToParentTransformProperty()**
- public Bounds **localToScene**(javafx.geometry.Bounds arg0)
- public Bounds **localToScene**(javafx.geometry.Bounds arg0, boolean arg1)
- public Point2D **localToScene**(double arg0, double arg1)
- public Point2D **localToScene**(double arg0, double arg1, boolean arg2)
- public Point3D **localToScene**(double arg0, double arg1, double arg2)
- public Point3D **localToScene**(double arg0, double arg1, double arg2, boolean arg3)
- public Point2D **localToScene**(javafx.geometry.Point2D arg0)
- public Point2D **localToScene**(javafx.geometry.Point2D arg0, boolean arg1)
- public Point3D **localToScene**(javafx.geometry.Point3D arg0)
- public Point3D **localToScene**(javafx.geometry.Point3D arg0, boolean arg1)
- public final ReadOnlyObjectProperty **localToSceneTransformProperty()**
- public Bounds **localToScreen**(javafx.geometry.Bounds arg0)
- public Point2D **localToScreen**(double arg0, double arg1)
- public Point2D **localToScreen**(double arg0, double arg1, double arg2)
- public Point2D **localToScreen**(javafx.geometry.Point2D arg0)
- public Point2D **localToScreen**(javafx.geometry.Point3D arg0)
- public Node **lookup**(java.lang.String arg0)
- public Set **lookupAll**(java.lang.String arg0)
- public final BooleanProperty **managedProperty()**
- public double **maxHeight**(double arg0)
- public double **maxWidth**(double arg0)
- public double **minHeight**(double arg0)
- public double **minWidth**(double arg0)
- public final BooleanProperty **mouseTransparentProperty()**
- public final ObjectProperty **nodeOrientationProperty()**

- public final void **notifyAccessibleAttributeChanged**(AccessibleAttribute arg0)
- public final ObjectProperty **onContextMenuRequestedProperty**()
- public final ObjectProperty **onDragDetectedProperty**()
- public final ObjectProperty **onDragDoneProperty**()
- public final ObjectProperty **onDragDroppedProperty**()
- public final ObjectProperty **onDragEnteredProperty**()
- public final ObjectProperty **onDragExitedProperty**()
- public final ObjectProperty **onDragOverProperty**()
- public final ObjectProperty **onInputMethodTextChangedProperty**()
- public final ObjectProperty **onKeyPressedProperty**()
- public final ObjectProperty **onKeyReleasedProperty**()
- public final ObjectProperty **onKeyTypedProperty**()
- public final ObjectProperty **onMouseClickedProperty**()
- public final ObjectProperty **onMouseDragEnteredProperty**()
- public final ObjectProperty **onMouseDragExitedProperty**()
- public final ObjectProperty **onMouseDraggedProperty**()
- public final ObjectProperty **onMouseDragOverProperty**()
- public final ObjectProperty **onMouseDragReleasedProperty**()
- public final ObjectProperty **onMouseEnteredProperty**()
- public final ObjectProperty **onMouseExitedProperty**()
- public final ObjectProperty **onMouseMovedProperty**()
- public final ObjectProperty **onMousePressedProperty**()
- public final ObjectProperty **onMouseReleasedProperty**()
- public final ObjectProperty **onRotateProperty**()
- public final ObjectProperty **onRotationFinishedProperty**()
- public final ObjectProperty **onRotationStartedProperty**()
- public final ObjectProperty **onScrollFinishedProperty**()
- public final ObjectProperty **onScrollProperty**()
- public final ObjectProperty **onScrollStartedProperty**()
- public final ObjectProperty **onSwipeDownProperty**()
- public final ObjectProperty **onSwipeLeftProperty**()
- public final ObjectProperty **onSwipeRightProperty**()
- public final ObjectProperty **onSwipeUpProperty**()
- public final ObjectProperty **onTouchMovedProperty**()
- public final ObjectProperty **onTouchPressedProperty**()
- public final ObjectProperty **onTouchReleasedProperty**()
- public final ObjectProperty **onTouchStationaryProperty**()
- public final ObjectProperty **onZoomFinishedProperty**()
- public final ObjectProperty **onZoomProperty**()
- public final ObjectProperty **onZoomStartedProperty**()
- public final DoubleProperty **opacityProperty**()
- public final ReadOnlyObjectProperty **parentProperty**()
- public Bounds **parentToLocal**(javafx.geometry.Bounds arg0)
- public Point2D **parentToLocal**(double arg0, double arg1)
- public Point3D **parentToLocal**(double arg0, double arg1, double arg2)
- public Point2D **parentToLocal**(javafx.geometry.Point2D arg0)
- public Point3D **parentToLocal**(javafx.geometry.Point3D arg0)
- public final BooleanProperty **pickOnBoundsProperty**()

- public double **prefHeight**(double arg0)
- public double **prefWidth**(double arg0)
- public final ReadOnlyBooleanProperty **pressedProperty**()
- public final void **pseudoClassStateChanged**(javafx.css.PseudoClass arg0, boolean arg1)
- public Object **queryAccessibleAttribute**(AccessibleAttribute arg0, java.lang.Object[] arg1)
- public void **relocate**(double arg0, double arg1)
- public final void **removeEventFilter**(javafx.event.EventType arg0, javafx.event.EventHandler arg1)
- public final void **removeEventHandler**(javafx.event.EventType arg0, javafx.event.EventHandler arg1)
- public void **requestFocus**()
- public void **resize**(double arg0, double arg1)
- public void **resizeRelocate**(double arg0, double arg1, double arg2, double arg3)
- public final DoubleProperty **rotateProperty**()
- public final ObjectProperty **rotationAxisProperty**()
- public final DoubleProperty **scaleXProperty**()
- public final DoubleProperty **scaleYProperty**()
- public final DoubleProperty **scaleZProperty**()
- public final ReadOnlyObjectProperty **sceneProperty**()
- public Bounds **sceneToLocal**(javafx.geometry.Bounds arg0)
- public Bounds **sceneToLocal**(javafx.geometry.Bounds arg0, boolean arg1)
- public Point2D **sceneToLocal**(double arg0, double arg1)
- public Point2D **sceneToLocal**(double arg0, double arg1, boolean arg2)
- public Point3D **sceneToLocal**(double arg0, double arg1, double arg2)
- public Point2D **sceneToLocal**(javafx.geometry.Point2D arg0)
- public Point2D **sceneToLocal**(javafx.geometry.Point2D arg0, boolean arg1)
- public Point3D **sceneToLocal**(javafx.geometry.Point3D arg0)
- public Bounds **screenToLocal**(javafx.geometry.Bounds arg0)
- public Point2D **screenToLocal**(double arg0, double arg1)
- public Point2D **screenToLocal**(javafx.geometry.Point2D arg0)
- public final void **setAccessibleHelp**(java.lang.String arg0)
- public final void **setAccessibleRole**(AccessibleRole arg0)
- public final void **setAccessibleRoleDescription**(java.lang.String arg0)
- public final void **setAccessibleText**(java.lang.String arg0)
- public final void **setBlendMode**(effect.BlendMode arg0)
- public final void **setCache**(boolean arg0)
- public final void **setCacheHint**(CacheHint arg0)
- public final void **setClip**(Node arg0)
- public final void **setCursor**(Cursor arg0)
- public final void **setDepthTest**(DepthTest arg0)
- public final void **setDisable**(boolean arg0)
- protected final void **setDisabled**(boolean arg0)
- public final void **setEffect**(effect.Effect arg0)

- `public final void setEventDispatcher(javafx.event.EventDispatcher arg0)`
- `protected final void setEventHandler(javafx.event.EventType arg0, javafx.event.EventHandler arg1)`
- `protected final void setFocused(boolean arg0)`
- `public final void setFocusTraversable(boolean arg0)`
- `protected final void setHover(boolean arg0)`
- `public final void setId(java.lang.String arg0)`
- `public final void setInputMethodRequests(input.InputMethodRequests arg0)`
- `public final void setLayoutX(double arg0)`
- `public final void setLayoutY(double arg0)`
- `public final void setManaged(boolean arg0)`
- `public final void setMouseTransparent(boolean arg0)`
- `public final void setNodeOrientation(javafx.geometry.NodeOrientation arg0)`
- `public final void setOnContextMenuRequested(javafx.event.EventHandler arg0)`
- `public final void setOnDragDetected(javafx.event.EventHandler arg0)`
- `public final void setOnDragDone(javafx.event.EventHandler arg0)`
- `public final void setOnDragDropped(javafx.event.EventHandler arg0)`
- `public final void setOnDragEntered(javafx.event.EventHandler arg0)`
- `public final void setOnDragExited(javafx.event.EventHandler arg0)`
- `public final void setOnDragOver(javafx.event.EventHandler arg0)`
- `public final void setOnInputMethodTextChanged(javafx.event.EventHandler arg0)`
- `public final void setOnKeyPressed(javafx.event.EventHandler arg0)`
- `public final void setOnKeyReleased(javafx.event.EventHandler arg0)`
- `public final void setOnKeyTyped(javafx.event.EventHandler arg0)`
- `public final void setOnMouseClicked(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragEntered(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragExited(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragged(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragOver(javafx.event.EventHandler arg0)`
- `public final void setOnMouseDragReleased(javafx.event.EventHandler arg0)`

- `public final void setOnMouseEntered(javafx.event.EventHandler arg0)`
- `public final void setOnMouseExited(javafx.event.EventHandler arg0)`
- `public final void setOnMouseMoved(javafx.event.EventHandler arg0)`
- `public final void setOnMousePressed(javafx.event.EventHandler arg0)`
- `public final void setOnMouseReleased(javafx.event.EventHandler arg0)`
- `public final void setOnRotate(javafx.event.EventHandler arg0)`
- `public final void setOnRotationFinished(javafx.event.EventHandler arg0)`
- `public final void setOnRotationStarted(javafx.event.EventHandler arg0)`
- `public final void setOnScroll(javafx.event.EventHandler arg0)`
- `public final void setOnScrollFinished(javafx.event.EventHandler arg0)`
- `public final void setOnScrollStarted(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeDown(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeLeft(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeRight(javafx.event.EventHandler arg0)`
- `public final void setOnSwipeUp(javafx.event.EventHandler arg0)`
- `public final void setOnTouchMoved(javafx.event.EventHandler arg0)`
- `public final void setOnTouchPressed(javafx.event.EventHandler arg0)`
- `public final void setOnTouchReleased(javafx.event.EventHandler arg0)`
- `public final void setOnTouchStationary(javafx.event.EventHandler arg0)`
- `public final void setOnZoom(javafx.event.EventHandler arg0)`
- `public final void setOnZoomFinished(javafx.event.EventHandler arg0)`
- `public final void setOnZoomStarted(javafx.event.EventHandler arg0)`
- `public final void setOpacity(double arg0)`
- `public final void setPickOnBounds(boolean arg0)`
- `protected final void setPressed(boolean arg0)`
- `public final void setRotate(double arg0)`
- `public final void setRotationAxis(javafx.geometry.Point3D arg0)`
- `public final void setScaleX(double arg0)`
- `public final void setScaleY(double arg0)`
- `public final void setScaleZ(double arg0)`
- `public final void setStyle(java.lang.String arg0)`

- `public final void setTranslateX(double arg0)`
- `public final void setTranslateY(double arg0)`
- `public final void setTranslateZ(double arg0)`
- `public void setUserData(java.lang.Object arg0)`
- `public final void setVisible(boolean arg0)`
- `public void snapshot(javafx.util.Callback arg0, SnapshotParameters arg1, image.WritableImage arg2)`
- `public WritableImage snapshot(SnapshotParameters arg0, image.WritableImage arg1)`
- `public Dragboard startDragAndDrop(input.TransferMode[] arg0)`
- `public void startFullDrag()`
- `public final StringProperty styleProperty()`
- `public void toBack()`
- `public void toFront()`
- `public String toString()`
- `public final DoubleProperty translateXProperty()`
- `public final DoubleProperty translateYProperty()`
- `public final DoubleProperty translateZProperty()`
- `public boolean usesMirroring()`
- `public final BooleanProperty visibleProperty()`

3.8 Class VueGraphiqueAideur

Contient les mthodes permettant d’afficher les lments dans la zone graphique

3.8.1 Declaration

```
public class VueGraphiqueAideur
    extends java.lang.Object
```

3.8.2 Constructor summary

[VueGraphiqueAideur\(StackPane, Group, ScrollPane, Slider\)](#) Constructeur de la vue graphique

3.8.3 Method summary

[afficherDemande\(\)](#) Affiche les livraisons contenues dans une demande de livraison

[afficherPlan\(\)](#) Affiche tous les points du plan et met jour la taille du canvas graphique.

[afficherTournee\(\)](#) Affiche la tourne (les tronons empreints entre les lieux de livraisons)

[construireDemande\(Demande\)](#) Stocke la demande

[construireGraphe\(PlanDeVille\)](#) Construit et affiche le graphe du plan de la ville sur le canvas graphique de la fentre

[construireTournee\(List\)](#) Construit et affiche la tourne

desactiverSurbrillance() Dsactive la surbrillance pour toutes les surbrillances

getCanvas()

nettoyerAffichage() Supprime tous les lments sur la partie graphique

setControleurApplication(ControleurInterface) Met jour le controleur de l'application pour la vue graphique

surbrillanceLivraison(Livraison) Affiche une livraison en surbrillance sur la partie graphique

surbrillanceLivraisons(Collection) Mets plusieurs livraisons en surbrillance sur la partie graphique

3.8.4 Constructors

- **VueGraphiqueAideur**

```
public VueGraphiqueAideur (javafx.scene.layout.
    StackPane canvas , javafx.scene.Group group ,
    javafx.scene.control.ScrollPane scrollPane ,
    javafx.scene.control.Slider slider )
```

- **Description**

Constructeur de la vue graphique

- **Parameters**

- * **canvas** – Le canvas sur lequel on dessinera les lments graphiques
- * **group** – Le group de la partie graphique
- * **scrollPane** – La barre de dfilement
- * **slider** – Le slide de zoom

3.8.5 Methods

- **afficherDemande**

```
public void afficherDemande ()
```

- **Description**

Affiche les livraisons contenues dans une demande de livraison

- **afficherPlan**

```
public void afficherPlan ()
```

- **Description**

Affiche tous les points du plan et met à jour la taille du canvas graphique. Les points sont toujours affichés par rapport : (leur taille initiale dans le fichier XML / la plus grande taille dans le fichier XML) => (la nouvelle taille / la taille du canvas)

- **afficherTournee**

```
public void afficherTournee()
```

- **Description**

Affiche la tournée (les tronçons empruntés entre les lieux de livraisons)

- **construireDemande**

```
public void construireDemande(modele.donneesxml.  
    Demande demande)
```

- **Description**

Stocke la demande

- **Parameters**

* **demande** – La demande de livraison

- **construireGraphe**

```
public void construireGraphe(modele.donneesxml.  
    PlanDeVille plan)
```

- **Description**

Construit et affiche le graphe du plan de la ville sur le canvas graphique de la fenêtre

- **Parameters**

* **plan** – Le plan de la ville, chargé par le contrôleur et persistant

- **construireTournee**

```
public void construireTournee(java.util.List  
    tournée)
```


- **Description**
Construit et affiche la tourne
- **Parameters**
* `tournee` – La tourne calcule

- **desactiverSurbrillance**

```
public void desactiverSurbrillance()
```

- **Description**
Dsactive la surbrillance pour toutes les surbrillances

- **getCanvas**

```
public javafx.scene.layout.StackPane getCanvas()
```

- **Returns** – Retourne la partie graphique pour dessin

- **nettoyerAffichage**

```
public void nettoyerAffichage()
```

- **Description**
Supprime tous les lments sur la partie graphique

- **setControleurApplication**

```
public void setControleurApplication(controleur.  
    ControleurInterface controleurApplication)
```

- **Description**
Met jour le controleur de l’application pour la vue graphique

- **surbrillanceLivraison**

```
public void surbrillanceLivraison(modele.  
    donneesxml.Livraison livraison)
```

- **Description**
Affiche une livraison en surbrillance sur la partie graphique

- **Parameters**

- * livraison – La livraison mettre en surbrillance

- **surbrillanceLivraisons**

```
public void surbrillanceLivraisons(java.util.  
Collection livraisons)
```

- **Description**

- Mets plusieurs livraisons en surbrillance sur la partie graphique

- **Parameters**

- * livraisons – Les livraisons mettre en surbrillance

3.9 Class VuePrincipale

Cette classe joue le rôle de binding pour la fenêtre principale de l'application. C'est ici qu'on spécifiera les couteurs et consorts. Remarque : Les couteurs peuvent être spécifiés directement dans le fichier xml aussi

3.9.1 Declaration

```
public class VuePrincipale  
extends java.lang.Object implements javafx.fxml.  
    Initializable, controleur.observeur.  
    ActivationOuvrirDemandeObservateur, controleur.  
    observeur.ActivationOuvrirPlanObservateur,  
    controleur.observeur.ModeleObservateur,  
    controleur.observeur.AnnulerCommandeObservateur,  
    controleur.observeur.  
    RetablirCommandeObservateur, controleur.  
    observeur.PlanChargeObservateur
```

3.9.2 Constructor summary

[VuePrincipale\(\)](#)

3.9.3 Method summary

[getAideurVueGraphique\(\)](#)

[initialiserObserveurs\(\)](#) Initialise les différents observateurs de
la vue principale

[initialize\(URL, ResourceBundle\)](#)

[notifierObservateurAnnulerCommande\(boolean\)](#)

```

notifierObservateurOuvrirDemande(boolean)
notifierObservateurRetablirCommande(boolean)
notifierObservateursModele()
notifierObservateursOuvrirPlan(boolean)
notifierObservateursPlanCharge()
setControleurApplication(ControleurInterface) Met jour
    le controleur de l'application
setVueGraphiqueControleurApplication(ControleurInterface)
    + Met jour le controleur de l'application pour la vue
    graphique

```

3.9.4 Constructors

- **VuePrincipale**

```

public VuePrincipale()

```

3.9.5 Methods

- **getAideurVueGraphique**

```

protected VueGraphiqueAideur getAideurVueGraphique
    ()

```

– **Returns** – La vue graphique

- **initialiserObserveurs**

```

protected void initialiserObserveurs()

```

– **Description**

Initialise les differents obserserveurs de la vue principale

- **initialize**

```

void initialize(java.net.URL arg0,java.util.
    ResourceBundle arg1)

```

- **notifierObservateurAnnulerCommande**

```

void notifierObservateurAnnulerCommande(boolean
    active)

```

- **Description** copied from **controlleur.observateur.AnnulerCommandeObservateur**
Notifie l'observateur s'il faut activer l'action dans le menu qui permet d'annuler une commande
- **Parameters**
 - * **active** – Dit si l'action Annuler doit etre active ou non

- **notifierObservateurOuvrirDemande**

```
void notifierObservateurOuvrirDemande( boolean
    activer )
```

- **Description** copied from **controlleur.observateur.ActivationOuvrirDemandeObservateur**
Notifie les observateurs qu'il faut ou pas activer l'element du menu qui permet d'ouvrir une demande
- **Parameters**
 - * **activer** – Vrai s'il faut s'activer ou se dsactiver lors de cette modification

- **notifierObservateurRetablirCommande**

```
void notifierObservateurRetablirCommande( boolean
    active )
```

- **Description** copied from **controlleur.observateur.RetablirCommandeObservateur**
Notifie l'observateur s'il faut activer l'action dans le menu qui permet de retablir une commande une commande
- **Parameters**
 - * **active** – Vrai s'il faut activer suite au rtablissement de la commande

- **notifierObservateursModele**

```
void notifierObservateursModele()
```

- **Description** copied from **controlleur.observateur.ModeleObservateur**
Notifie les observeurs que le modle a chang

- **notifierObservateursOuvrirPlan**

```
void notifierObservateursOuvrirPlan( boolean
    activer )
```

- **Description** copied from **controlleur.observateur.ActivationOuvrirPlanObservateur**

Notifie les observateurs qu'il faut ou pas activer l'element du menu qui permet d'ouvrir un plan

- **Parameters**

* **activer** – Le plan a t charger ou dsactiv

- **notifierObservateursPlanCharge**

```
void notifierObservateursPlanCharge()
```

- **Description** copied from **controlleur.observateur.PlanChargeObservateur**

Notifie la vue qu'un nouveau plan a t charg.

- **setControleurApplication**

```
public void setControleurApplication( controleur .
    ControleurInterface controleurApplication )
```

- **Description**

Met jour le controleur de l'application

- **Parameters**

* **controleurApplication** – Le nouveau controleur d'interface

- **setVueGraphiqueControleurApplication**

```
public void setVueGraphiqueControleurApplication (
    controleur . ControleurInterface
    controleurApplication )
```

- **Description**

+ Met jour le controleur de l'application pour la vue graphique

3.10 Class VueTextuelle

Cette classe gère les livraisons et leurs horaires. Elle s'occupe de la vue textuelle qui se trouve à gauche dans la fenêtre principale.

3.10.1 Declaration

```
public class VueTextuelle
    extends java.lang.Object implements javafx.fxml.
        Initializable, controleur.observeur.
        ModeleObserveur, controleur.observeur.
        PlanChargeObserveur
```

3.10.2 Constructor summary

[VueTextuelle\(\)](#)

3.10.3 Method summary

[construireVueTableLivraison\(Demande\)](#) Construit la table des livraisons

[initialiserObserveurs\(\)](#) Ajoute la vue textuelle comme observeurs au près du contrôleur

[initialize\(URL, ResourceBundle\)](#) Méthode appelée automatiquement au chargement du fichier XML

[notifierObserveursModele\(\)](#) Notification déclenche lors d'un changement dans le modèle.

[notifierObserveursPlanCharge\(\)](#) Efface le contenu de la table à chaque chargement d'un plan.

[setAideurVueGraphique\(VueGraphiqueAideur\)](#) Met à jour la référence vers la vue graphique

[setContrôleurApplication\(ContrôleurInterface\)](#) Met à jour le contrôleur de l'application pour la vue textuelle

3.10.4 Constructors

- [VueTextuelle](#)

```
public VueTextuelle()
```

3.10.5 Methods

- [construireVueTableLivraison](#)

protected void construireVueTableLivraison (modele.
donneesxml.Demande demande)

– **Description**

Contruit la table des livraisons

– **Parameters**

* **demande** – La demande de livraison charge partir d’un fichier XML. La demande ne doit pas tre null

• **initialiserObserveurs**

protected void initialiserObserveurs ()

– **Description**

Ajoute la vue textuelle comme observeurs au prs du controleur

• **initialize**

public void initialize (java.net.URL location , java.
util.ResourceBundle resources)

– **Description**

Mthode appele automatiquement au chargement du fichier XML

• **notifierObservateursModele**

public void notifierObservateursModele ()

– **Description**

Notification dclenche lors d’un changement dans le model. Cette peut etre notification est dclenche que si on a dja charg un plan et une demande de livraison dans l’application

• **notifierObservateursPlanCharge**

public void notifierObservateursPlanCharge ()

– **Description**

Efface le contenu de la table chaque chargement d’un plan.

- **setAideurVueGraphique**

```
protected void setAideurVueGraphique(  
    VueGraphiqueAideur vueGraphique)
```

- **Description**

Met à jour la référence vers la vue graphique

- **setContrôleurApplication**

```
protected void setContrôleurApplication(contrôleur  
    .ContrôleurInterface contrôleurApplication)
```

- **Description**

Met à jour le contrôleur de l'application pour la vue textuelle

- **Parameters**

- * **contrôleurApplication** – contrôleur initialisé (non null)

Chapter 4

Package

<i>Package Contents</i>	<i>Page</i>
Classes	
DevOO	2
Classe d'entre de l'application	

4.1 Class DevOO

Classe d'entre de l'application

4.1.1 Declaration

```
public class DevOO
    extends java.lang.Object
```

4.1.2 Constructor summary

[DevOO\(\)](#)

4.1.3 Method summary

[main\(String\[\]\)](#) Point d'entre de l'application

4.1.4 Constructors

- DevOO

```
public DevOO()
```

4.1.5 Methods

- **main**

```
public static void main(java.lang.String [] args)
```

- **Description**

Point d'entre de l'application

- **Parameters**

- * **args** – Les arguments en ligne de commande