



DALMAYRAC Gorka  
CAMPISTRON Julian

# **Dossier de conception Lecteur de Diaporama v2**

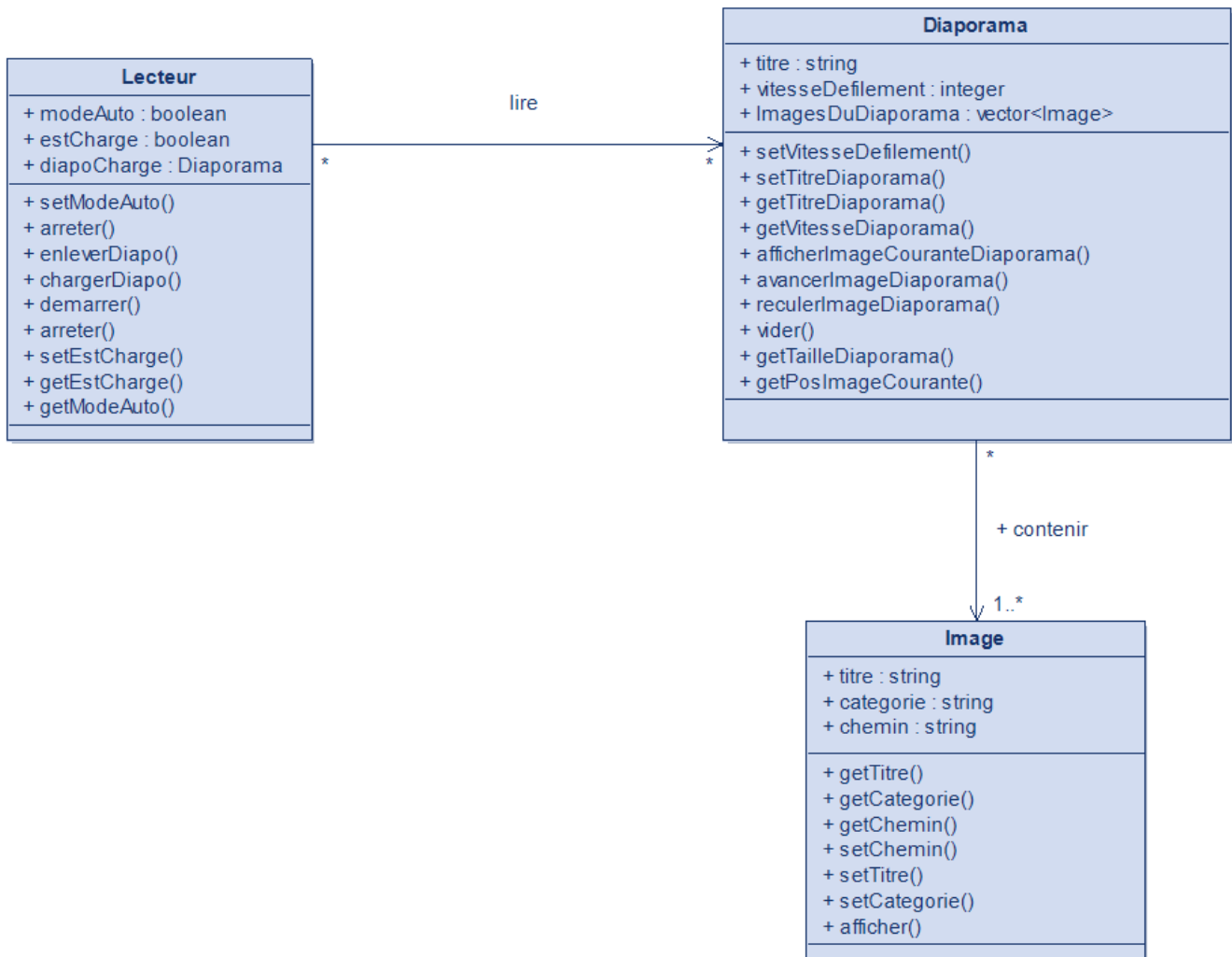
**Groupe III-TP5-4**  
**SAE S2.01 Développement d'une  
application**

**Date de remise : 29/04/2024**

## Table des matières

UML : diagramme des classes.....	3
Documentation des méthodes et attributs de chaque classe .....	4
1. Classe Lecteur .....	4
Attributs.....	4
Méthodes.....	4
2. Classe Diaporama .....	5
Attributs.....	5
Méthodes.....	5
3. Classe Image.....	6
Attributs.....	6
Méthodes.....	6
Diagrammes état-transition de l'application .....	7
4. Forme UML.....	7
5. Forme matricielle .....	8
Lien Eléments de l'interface – Fonctionnalités .....	8

## UML : diagramme des classes



# Documentation des méthodes et attributs de chaque classe

## 1. Classe Lecteur

### Signification des attributs et méthodes de la classe Lecteur

La classe Lecteur est conçue pour gérer l'affichage des diapositives. Elle possède plusieurs attributs et méthodes :

### Attributs

- 1. Diaporama diapoCharge** : Cet attribut est de type Diaporama et représente le diaporama actuellement chargé dans le lecteur. Il est utilisé pour charger différents diaporamas.
- 2. bool modeAuto** : Cet attribut est de type bool et représente l'état du défilement automatique des diapositives. Si modeAuto est true, cela signifie que les diapositives sont défilées automatiquement sans intervention de l'utilisateur.
- 3. bool estCharge** : Cet attribut est également de type bool et représente l'état de chargement d'un diaporama. Si estCharge est true, cela signifie qu'un diaporama est actuellement chargé dans le lecteur.

### Méthodes

- 1. Lecteur::Lecteur(Diaporama diapoCharge, bool modeAuto, bool estCharge)** : Cette méthode est le constructeur de la classe Lecteur. Elle initialise les attributs diapoCharge, modeAuto, et estCharge avec les valeurs passées en paramètre.
- 2. ~Lecteur()** : Cette méthode est le destructeur de la classe Lecteur. Elle libère les ressources allouées lors de la création de l'objet Lecteur.
- 3. void chargerDiapo(Diaporama diapoCharge)** : Cette méthode permet de charger une nouvelle diapositive dans le lecteur. Elle prend en paramètre un objet Diaporama et charge la diapositive associée.
- 4. void enleverDiapo()** : Cette méthode permet de retirer la diapositive actuellement chargée dans le lecteur. Elle ne prend pas de paramètre et ne retourne rien.
- 5. void demarrer()** : Cette méthode permet de démarrer l'affichage automatique des diapositives. Elle ne prend pas de paramètre et ne retourne rien.
- 6. void arreter()** : Cette méthode permet de stopper l'affichage automatique des diapositives. Elle ne prend pas de paramètre et ne retourne rien.
- 7. void setModeAuto(bool modeAuto)** : Cette méthode permet de modifier l'état de l'affichage automatique des diapositives. Elle prend en paramètre un booléen modeAuto et modifie l'état de l'affichage automatique des diapositives en conséquence.

**8. void setEstCharge(bool estCharge) :** Cette méthode permet de modifier l'état de chargement d'une diapositive. Elle prend en paramètre un booléen estCharge et modifie l'état de chargement d'une diapositive en conséquence.

**9. bool getMode() :** Cette méthode permet de récupérer l'état de l'affichage automatique des diapositives. Elle ne prend pas de paramètre et retourne un booléen représentant l'état de l'affichage automatique des diapositives.

**10. bool getEstCharge() :** Cette méthode permet de récupérer l'état de chargement d'une diapositive. Elle ne prend pas de paramètre et retourne un booléen représentant l'état de chargement d'une diapositive.

## 2. Classe Diaporama

La classe Diaporama est conçue pour gérer les diaporamas.

### Attributs

**1. string titreDiaporama :** Cet attribut est de type string et représente le titre du diaporama. Il est utilisé pour afficher le titre du diaporama.

**2. unsigned int vitesseDefilement :** Cet attribut est de type unsigned int et représente la vitesse de défilement des images dans le diaporama. Il est utilisé pour contrôler la vitesse à laquelle les images sont affichées.

**3. vector<Image> ImagesDuDiaporama :** Cet attribut est un vecteur d'images de type Image et représente l'ensemble des images comprises dans le diaporama. Il est utilisé pour afficher l'image souhaitée du diaporama ou bien encore modifier celui-ci

**4. int posImageCourante :** Cet attribut est de type unsigned int et représente la position du « curseur » dans le vecteur des images du diaporama. Il est donc initialisé à 0, et n'est pas présent dans les constructeurs. Il permet de parcourir le vecteur des images du diaporama

### Méthodes

**1. Diaporama(string pTitre, unsigned int pVitesseDefilement) :** Cette méthode est le constructeur de la classe Diaporama. Elle prend en paramètre un titre pTitre et une vitesse de défilement pVitesseDefilement et initialise les attributs titreDiaporama et vitesseDefilement en conséquence.

**2. void vider() :** Cette méthode permet de vider le diaporama. Elle supprime toutes les images du diaporama.

**3. void setVitesseDefilement(unsigned int pVitesseDefilement) :** Cette méthode permet de modifier la vitesse de défilement du diaporama. Elle prend en paramètre une nouvelle vitesse pVitesseDefilement et modifie l'attribut vitesseDefilement en conséquence.

**4. void setTitreDiaporama(string pTitre) :** Cette méthode permet de modifier le titre du diaporama. Elle prend en paramètre un nouveau titre pTitre et modifie l'attribut titreDiaporama en conséquence.

**5. string getTitreDiaporama() :** Cette méthode permet de récupérer le titre du diaporama. Elle ne prend pas de paramètre et retourne une chaîne de caractères représentant le titre du diaporama.

**6. int getVitesseDefilement()** : Cette méthode permet de récupérer la vitesse de défilement du diaporama. Elle ne prend pas de paramètre et retourne un entier représentant la vitesse de défilement du diaporama.

**7. int getTailleDiaporama()** : Cette méthode permet de récupérer la taille du diaporama. Elle ne prend pas de paramètre et retourne un entier représentant le nombre d'images dans le diaporama.

**8. int getPosImageCourante()** : Cette méthode permet de récupérer la position de l'image courante dans le diaporama. Elle ne prend pas de paramètre et retourne un entier représentant la position de l'image courante dans le diaporama.

**9. void ajouterImage(Image image)** : Cette méthode permet d'ajouter une image au diaporama. Elle prend en paramètre une image image et ajoute cette image à la fin du diaporama.

**10. void afficherImageCouranteDiaporama()** : Cette méthode permet d'afficher l'image courante du diaporama. Elle ne prend pas de paramètre et affiche l'image à la position posImageCourante dans le diaporama

**11. void avancerImageDiaporama()** : Cette méthode permet de passer à l'image suivante du diaporama. Elle ne prend pas de paramètre et incrémente la variable posImageCourante.

**12. void reculerImageDiaporama()** : Cette méthode permet de passer à l'image précédente du diaporama. Elle ne prend pas de paramètre et décrémente la variable posImageCourante.

### 3. Classe Image

La classe Image est conçue pour gérer les images. Elle possède plusieurs attributs et méthodes qui sont décrits ci-dessous.

#### Attributs

**1. string titre** : Cet attribut est de type string et représente le titre de l'image.

**2. string categorie** : Cet attribut est de type string et représente la catégorie de l'image. Il est utilisé pour fournir des informations supplémentaires sur l'image.

**3. string chemin** : Cet attribut est de type string et représente le chemin d'accès de l'image. Il est utilisé pour spécifier l'emplacement de l'image sur le disque dur.

#### Méthodes

**1. Image(string pTitre, string pCategorie, string pChemin)** : Cette méthode est le constructeur de la classe Image. Elle prend en paramètre un titre pTitre, une categorie pCategorie et un chemin pChemin et initialise les attributs titre, categorie et chemin en conséquence.

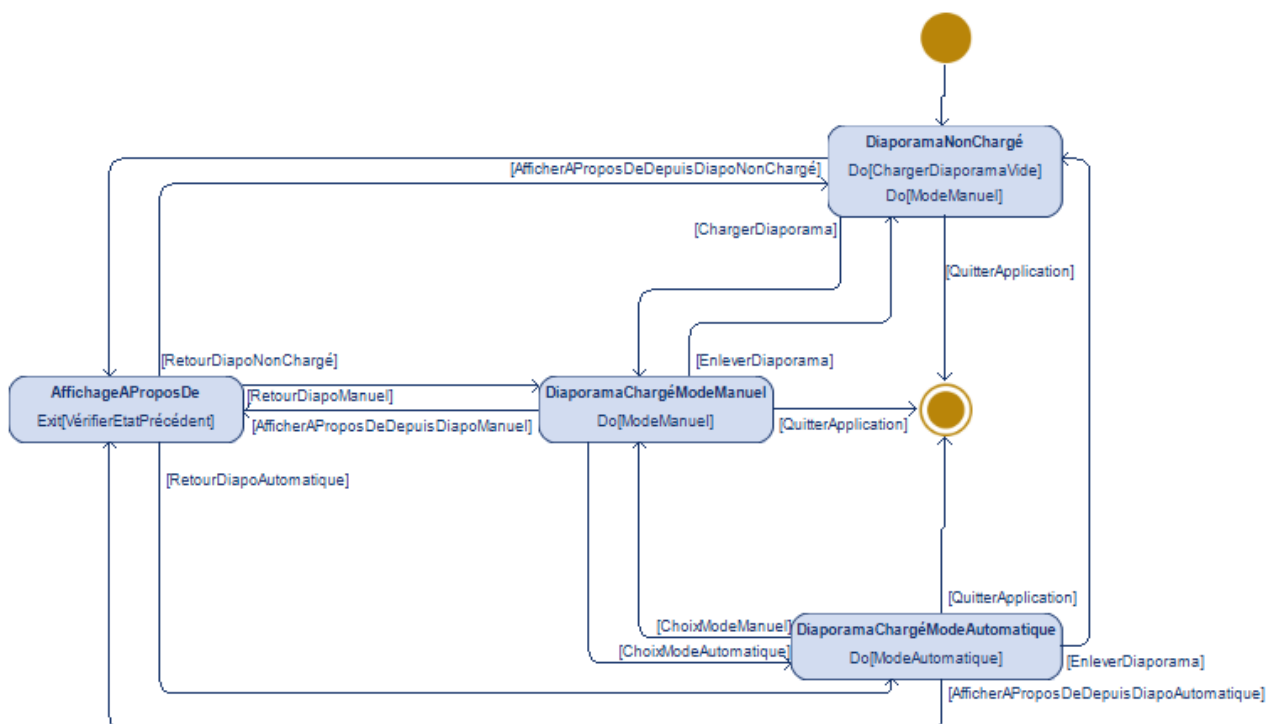
**2. void afficher()** : Cette méthode permet d'afficher l'image. Elle ne prend pas de paramètre et affiche l'image à l'écran.

**3. void setTitre(string pTitre)** : Cette méthode permet de modifier le titre de l'image. Elle prend en paramètre un nouveau titre pTitre et modifie l'attribut titre en conséquence.

4. **void setCategorie(string pCategorie)** : Cette méthode permet de modifier la catégorie de l'image. Elle prend en paramètre une nouvelle catégorie pCategorie et modifie l'attribut categorie en conséquence.
5. **void setChemin(string pChemin)** : Cette méthode permet de modifier le chemin d'accès de l'image. Elle prend en paramètre un nouveau chemin pChemin et modifie l'attribut chemin en conséquence.
6. **string getTitre()** : Cette méthode permet de récupérer le titre de l'image. Elle ne prend pas de paramètre et retourne une chaîne de caractères représentant le titre de l'image.
7. **string getCategorie()** : Cette méthode permet de récupérer la catégorie de l'image. Elle ne prend pas de paramètre et retourne une chaîne de caractères représentant la catégorie de l'image.
8. **string getChemin()** : Cette méthode permet de récupérer le chemin d'accès de l'image. Elle ne prend pas de paramètre et retourne une chaîne de caractères représentant le chemin d'accès de l'image.

## Diagrammes état-transition de l'application

### 4. Forme UML



## 5. Forme matricielle

		actionA_propos_de	boutonValider_A_propos_de	actionCharger_diaporama	actionEnlever_diaporama	boutonLancer_diaporama	boutonArreter_diaporama	actionQuitter
Evénements	Etats	AfficherAProposDe	RetirerAProposDe	ChargerDiaporama	EnleverDiaporama	BasculerModeAutomatique	BasculerModeManuel	QuitterApplication
DiaporamaNonChargé	AffichageAProposDe	---	---	DiaporamaManuel	---	---	---	---
DiaporamaManuel	AffichageAProposDe	---	---	DiaporamaManuel	---	DiaporamaAutomatique	---	---
DiaporamaAutomatique	AffichageAProposDe	---	---	DiaporamaManuel	DiaporamaNonChargé	---	DiaporamaManuel	---
AffichageAProposDe	---	[EtatPrécédent=DiaporamaNonChargé]DiaporamaNonChargé	---	---	---	---	---	---
	---	[EtatPrécédent=DiaporamaManuel]DiaporamaManuel	---	---	---	---	---	---
	---	[EtatPrécédent=DiaporamaAutomatique]DiaporamaAutomatique	---	---	---	---	---	---

## Lien Eléments de l'interface – Fonctionnalités

**Label « Image »** (QLabel imgDiapo) : Label permettant l'affichage de l'image

**Bouton «Suivant »** (QPushButton btnSuivant) : Ce bouton déclenche le passage à l'image suivante du diaporama

**Bouton «Précédent »** (QPushButton btnPrecedent) : Ce bouton déclenche le passage à l'image précédente du diaporama

**Bouton «Lancer Diaporama »** (QPushButton btnLancer) : Ce bouton déclenche le passage au mode Automatique du diaporama, et débute donc le défilement d'images

**Bouton «Arrêter Diaporama »** (QPushButton btnArreter) : Ce bouton déclenche le passage au mode Manuel du diaporama

**Menu « Fichier »** (QMenu menuFichier)

Action « Quitter » : Permet la fermeture de l'application graphique

**Menu « Parametres »** (QMenu menuParametres) : Menu permettant l'accès aux divers paramètres du diaporama ci-dessous

Action « Charger Diaporama » : Permet de charger un diaporama dans le lecteur

Action « Enlever Diaporama » : Permet la suppression du diaporama présent dans le lecteur

Action « Vitesse de défilement » : Permet l'ajustement de la vitesse de défilement



Action « Filtrer » : Permet de filtrer les images du diaporama chargé dans le lecteur

**Menu « Aide »** (QMenu menuAide)

Action « A propos de... » : Permet d'obtenir des informations sur la version de l'application ainsi que ses auteurs