

S2.01 - Développement d'une application  
**Documentation des classes**

---



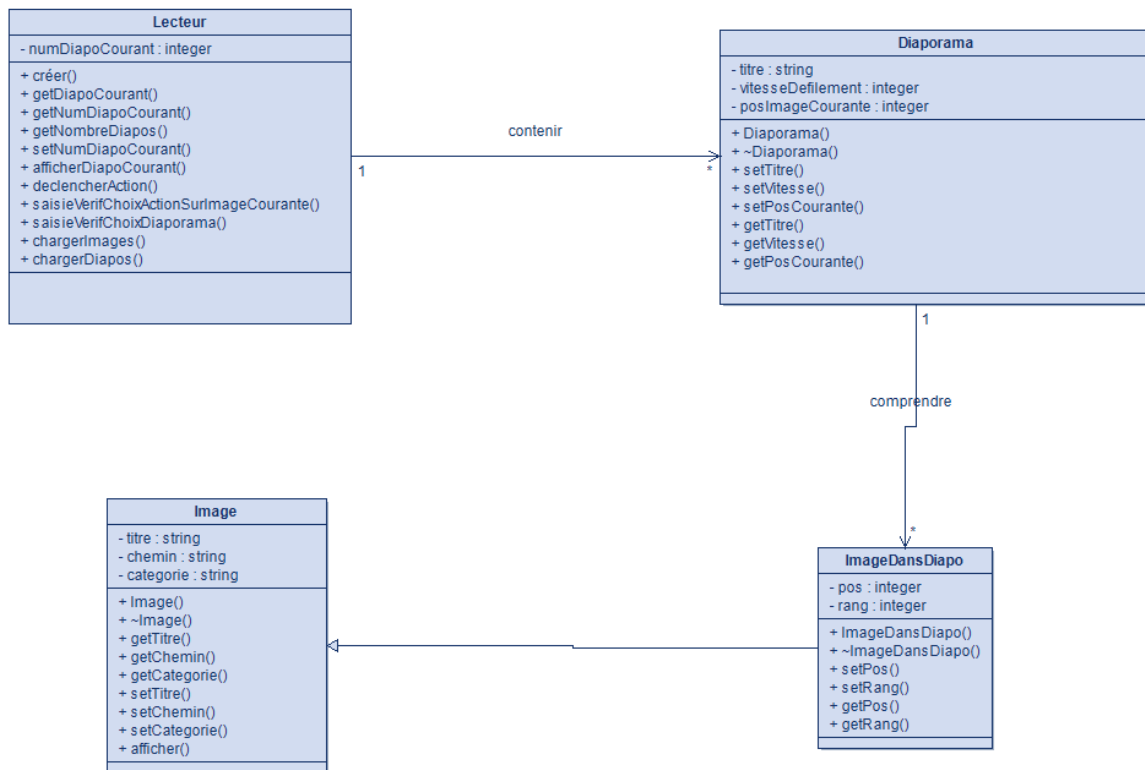
---

BARLIC François  
BOURCIEZ Maxime  
DUMAI Étienne  
TDII - TP3

## Table des matières :

I.	Diagramme de classes	3
II.	Type nécessaires déclarés	4
	<b>Diaporamas</b>	4
	<b>ImagesDuDiaporama</b>	4
	<b>Images</b>	4
VII.	Classe Lecteur	5
	<b>Attributs</b>	5
	<b>Méthodes publiques</b>	5
VIII.	Classe Diaporama	6
	<b>Attributs</b>	6
	<b>Méthodes publiques</b>	6
IX.	Classe ImageDansDiaporama	7
	<b>Héritage</b>	7
	<b>Attributs</b>	7
	<b>Méthodes publiques</b>	7
X.	Classe Image	8
	<b>Attributs</b>	8
	<b>Méthodes publiques</b>	8

# I. Diagramme de classes



**Figure 1 - Diagramme des classes du lecteur de diaporama**

Commentaires :

- L'association "contenir" se traduira dans le code par un attribut nommé "allDiapos" contenant tous les diaporamas du lecteur dans la classe Lecteur
- L'association "comprendre" se traduira dans le code par un attribut nommé "localisationImages" contenant les images du diaporama (classe ImageDansDiaporama) dans la classe Diaporama

## II. Type nécessaires déclarés

### Diaporamas

- **Description** : Alias pour un vecteur de `Diaporama`.
  - **Utilisation** : Permet de définir un type `Diaporamas` qui peut être utilisé pour stocker une collection de diaporamas.
  - **Déclaration** : `typedef vector<Diaporama> Diaporamas;`
  - **Exemple d'utilisation** : `Diaporamas mesDiaporamas;`
- 

### ImagesDuDiaporama

- **Description** : Alias pour un vecteur de `imageDansDiaporama`.
  - **Utilisation** : Permet de définir un type `ImagesDuDiaporama` qui peut être utilisé pour stocker une collection d'images dans un diaporama.
  - **Déclaration** : `typedef vector<imageDansDiaporama> ImagesDuDiaporama;`
  - **Exemple d'utilisation** : `ImagesDuDiaporama imagesDiaporama;`
- 

### Images

- III. **Description** : Alias pour un vecteur d'`Image`.
  - IV. **Utilisation** : Permet de définir un type `Images` qui peut être utilisé pour stocker une collection d'images.
  - V. **Déclaration** : `typedef vector<Image> Images;`
  - VI. **Exemple d'utilisation** : `Images mesImages;`
-

## VII. Classe Lecteur

### Attributs

- **Diaporamas \_allDiapos** : Un vecteur contenant tous les diaporamas gérés par le lecteur.<sup>1</sup>
- **unsigned int \_numDiapoCourant** : Le numéro du diaporama courant.

### Méthodes publiques

#### Constructeurs

- **Lecteur()** : Constructeur par défaut de la classe **Lecteur**.
- **Lecteur(Diaporamas diapos, unsigned int numDiapoCourant)** : Constructeur prenant en paramètres un vecteur de diaporamas et le numéro du diaporama courant.

#### Getters

- **Diaporamas getAllDiapos() const** : Retourne une copie du vecteur de tous les diaporamas.
- **Diaporama& getDiapoCourant()** : Retourne une référence vers le diaporama courant.
- **unsigned int getNumDiapoCourant() const** : Retourne le numéro du diaporama courant.
- **unsigned int getNombreDiapos() const** : Retourne le nombre total de diaporamas.

#### Setters

- **void setAllDiapos(const Diaporamas& vecteurDiapos)** : Définit le vecteur de tous les diaporamas.
- **void setNumDiapoCourant(unsigned int numDiapoCourant)** : Définit le numéro du diaporama courant.

#### Autres méthodes

- **void afficherDiapoCourant()** : Affiche les détails du diaporama courant.
- **void declencherAction(char pChoixAction)** : Déclenche une action spécifique sur le lecteur en fonction du choix de l'utilisateur.
- **void saisieVerifChoixActionSurImageCourante(char& pChoixAction)** : Saisie et vérifie le choix de l'utilisateur concernant les actions sur l'image courante.
- **unsigned int saisieVerifChoixDiaporama()** : Saisie et vérifie le choix de l'utilisateur pour changer de diaporama.
- **void charger(Images& images)** : Charge un ensemble d'images.
- **void chargerDiapos(Images images)** : Charge les diaporamas à partir d'un ensemble d'images.

## VIII. Classe Diaporama

### Attributs

- **string \_titre** : Le titre du diaporama.
- **unsigned int \_vitesseDefilement** : La vitesse de défilement du diaporama.
- **ImagesDuDiaporama \_localisationImages** : Un vecteur contenant les images du diaporama.
- **unsigned int \_posImageCourante** : La position de l'image courante dans le diaporama.

### Méthodes publiques

#### Constructeurs

- **Diaporama()** : Constructeur par défaut de la classe **Diaporama**.
- **Diaporama(string, unsigned int = 1, ImagesDuDiaporama = {}, unsigned int = 0)** : Constructeur permettant d'initialiser les attributs du diaporama.

#### Getters

- **string getTitre()const** : Retourne le titre du diaporama.
- **ImagesDuDiaporama getLocalisationImages() const** : Retourne une copie du vecteur contenant les images du diaporama.
- **unsigned int getVitesseDefilement() const** : Retourne la vitesse de défilement du diaporama.
- **unsigned int getNombreImages() const** : Retourne le nombre total d'images dans le diaporama.
- **unsigned int getPosImageCourante() const** : Retourne la position de l'image courante dans le diaporama.
- **imageDansDiaporama getImageCourante() const** : Retourne l'image courante du diaporama.

#### Setters

- **void setTitre(const string&)** : Définit le titre du diaporama.
- **void setVitesseDefilement(unsigned int)** : Définit la vitesse de défilement du diaporama.
- **void setLocalisationImages(const ImagesDuDiaporama&)** : Définit le vecteur contenant les images du diaporama.
- **void setPosImageCourante(unsigned int)** : Définit la position de l'image courante dans le diaporama.

#### Autres méthodes

- **void addImage(const imageDansDiaporama&)** : Ajoute une nouvelle image au diaporama.
- **void avancer()** : Fait avancer le diaporama en changeant l'image courante.
- **void reculer()** : Fait reculer le diaporama en changeant l'image courante.
- **void afficherImageCouranteDansDiaporamaCourant () const** : Affiche l'image courante dans le diaporama courant.
- **void triCroissantRang()** : Trie les images du diaporama par leur rang dans l'ordre croissant.

## IX. Classe ImageDansDiaporama

### Héritage

- **public Image** : Cette classe hérite publiquement de la classe **Image**.

### Attributs

- **unsigned int \_rang** : Le rang de l'image dans le tableau d'images.
- **unsigned int \_pos** : Le rang de l'image dans le diaporama.

### Méthodes publiques

#### Constructeurs

- **imageDansDiaporama()** : Constructeur par défaut de la classe **imageDansDiaporama**.
- **imageDansDiaporama(const Image&, unsigned int, unsigned int)** : Constructeur permettant d'initialiser les attributs de l'image dans le diaporama en utilisant une instance de la classe **Image**.
- **imageDansDiaporama(Images&, unsigned int, unsigned int)** : Constructeur permettant d'initialiser les attributs de l'image dans le diaporama en utilisant un vecteur d'images.

#### Getters

- **unsigned int getRang() const** : Retourne le rang de l'image dans le diaporama.
- **unsigned int getPos() const** : Retourne le rang de l'image dans le diaporama.
- **const Image& getImage() const** : Retourne une référence constante vers l'objet **Image** encapsulé.

#### Setters

- **void setRang(unsigned int)** : Définit le rang de l'image dans le diaporama.
- **void setPos(unsigned int)** : Définit le rang de l'image dans le diaporama.

#### Autres méthodes

- **void afficherImageCourante() const** : Affiche les informations de l'image courante dans le diaporama.

## X. Classe Image

### Attributs

- **string \_titre** : Intitulé de l'image.
- **string \_categorie** : Catégorie de l'image (personne, animal, objet).
- **string \_chemin** : Chemin d'accès à l'image.

### Méthodes publiques

#### Constructeurs

- **Image()** : Constructeur par défaut de la classe **Image**.
- **Image(const Image&)** : Constructeur de copie de la classe **Image**.
- **Image(string, string, string)** : Constructeur permettant d'initialiser tous les attributs de la classe **Image**.

#### Getters

- **string getCategorie() const** : Retourne la catégorie de l'image.
- **string getTitre() const** : Retourne le titre de l'image.
- **string getChemin() const** : Retourne le chemin d'accès à l'image.

#### Setters

- **void setCategorie(string)** : Définit la catégorie de l'image.
- **void setTitre(string)** : Définit le titre de l'image.
- **void setChemin(string)** : Définit le chemin d'accès à l'image.

#### Autres méthodes

- **void afficher() const** : Affiche les informations de l'image, y compris le titre, la catégorie et le chemin d'accès.