

BUT Informatique Semestre 2 2023-2024

IUT de Bayonne et du Pays Basque, Département Informatique

S2.01 « *Développement d’une application* »

**Lecteur de diaporamas – Dossier d’analyse et de conception**

ARANDIA Iban, TD2 – TP4

CHIPY Thibault, TD2- TP4

LATXAGUE Thibault, TD2 – TP4

Table des matières

[1. Version 1 – Orienté objet 3](#_Toc165729906)

[1.1. Diagramme des classes UML 3](#_Toc165729907)

[1.2. Attributs et méthodes des classes 4](#_Toc165729908)

[1.2.1 Classe Image 4](#_Toc165729909)

[1.2.1. Classe Lecteur 4](#_Toc165729910)

[1.2.3. Classe ImageDansDiaporama 6](#_Toc165729911)

[1.2.4. Classe Diaporama 7](#_Toc165729912)

[2. Version 2 - projet version graphique interface complète 9](#_Toc165729913)

[2.1. Diagramme état-transitions classique 9](#_Toc165729914)

[2.2. Diagramme état-transitions matriciel 9](#_Toc165729915)

[2.3. Liens entre éléments d’interface et fonctionnalités 10](#_Toc165729916)

# Version 1 – Orienté objet

## Diagramme des classes UML

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

## Attributs et méthodes des classes

### Classe Image

|  |  |  |  |
| --- | --- | --- | --- |
| Classe Image | | | |
| Nom attribut | Signification | Type | Exemple |
| m\_titre | Le titre de l’image | Chaine de caractères | Blanche Neige |
| m\_categorie | La catégorie de l’image | Chaine de caractères | Personnage |
| m\_chemin | Le chemin de l’image | Chaine de caractères | C:\\cartesDisney\\Disney\_4.gif |

|  |  |
| --- | --- |
| Nom méthode | Signification / but |
| Image | Constructeur par défaut de la classe Image avec comme valeur par défaut : « Def » |
| Image | Constructeur de copie |
| getCategorie() const | Getter pour récupérer la catégorie d’une image |
| getTitre() const | Getter pour récupérer le titre d’une image |
| getChemin() const | Getter pour récupérer le chemin d’accès d’une image |
| afficher() const | Procédure pour afficher une image, c’est-à-dire toutes ses caractéristiques : titre, catégorie et chemin d’accès |

### 1.2.1. Classe Lecteur

|  |  |  |  |
| --- | --- | --- | --- |
| Classe Lecteur | | | |
| Nom attribut | Signification | Type | Exemple |
| m\_toutesDiapos | Tableau dynamique de toutes les diapositives | Vector <Diaporama> | / |
| m\_numDiapoCourant | Le numéro du diaporama courant | Entier | 2 |
| m\_nombreDiapos | Le nombre de diaporama dans le lecteur | Entier | 5 |

|  |  |
| --- | --- |
| Nom méthode | Signification / but |
| Lecteur() | Constructeur par défaut de la classe Lecteur |
| Lecteur(vector<Diaporama>, unsigned int = 0) | Constructeur par copie de la classe Lecteur |
| vector<Diaporama> & getToutesDiapos() | Getter permettant d’obtenir tous les diaporamas. |
| Diaporama & getDiapoCourant() | Getter permettant d’obtenir le diaporama courant. |
| unsigned int getNumDiapoCourant() | Getter permettant d’obtenir le numéro du diaporama courant. |
| unsigned int getNombreDiapos() | Getter permettant d’obtenir le nombre de diaporamas. |
| Void setToutesDiapos(const vector<Diaporama>&) | Setter pour définir la liste complète des diaporamas à partir du tableau de diaporama fourni en paramètre. |
| Void setNumDiapoCourant(unsigned int) | Setter pour définir le numéro du diaporama courant à partir du nombre en paramètre. |
| Void setNombreDiapos(unsigned int) | Setter pour définir le nombre total de diapositives dans le diaporama à partir du nombre en paramètre. |
| Void declencherAction(char) | Procédure pour déclencher une action spécifique basée sur le caractère fourni en paramètre. |
| Void saisieVerifChoixActionSurImageCourante (char&) | Saisit et vérifie le choix de l'action à effectuer sur l'image courante du diaporama. |
| unsigned int saisieVerifChoixDiaporama() | Saisit et vérifie le choix du diaporama à afficher. |
| Void charger(Images&) | Procédure pour charger les images. |
| void chargerDiapos(Images&) | Procédure pour charger les diapositives pour le diaporama, utilisant les images chargées. |

### 1.2.3. Classe ImageDansDiaporama

|  |  |  |  |
| --- | --- | --- | --- |
| Classe ImageDansDiaporama | | | |
| Nom attribut | Signification | Type | Exemple |
| m\_pos | Rang de l’image dans le tableau d’image | Entier | 3 |
| m\_rang | Rang de l’image dans le diaporama | Entier | 1 |

|  |  |
| --- | --- |
| Nom méthode | Signification / but |
| ImageDansDiaporama() | Constructeur par défaut |
| ImageDansDiaporama(const Image&, unsigned int, unsigned int) | Constructeur par copie |
| ImageDansDiaporama(Images&, unsigned int, unsigned int) | Constructeur à partir d’une image |
| unsigned int getPosition() const | Getter pour obtenir la position d’une image dans un diaporama |
| unsigned int getRang() const | Getter pour obtenir le rang d’une image dans un diaporama |
| Image getImage() const | Getter pour obtenir l’image dans le diaporama |
| void setPosition(unsigned int) | Setter pour définir la position d’une image dans un diaporama |
| void setRang(unsigned int) | Setter pour définir le rang d’une image dans le diaporama |

### 1.2.4. Classe Diaporama

|  |  |  |  |
| --- | --- | --- | --- |
| Classe Diaporama | | | |
| Nom attribut | Signification | Type | Exemple |
| m\_titre | Le titre du diaporama | Chaine de caractères | Diaporama Yann |
| m\_vitesseDefilement | La vitesse de défilement des images (si mode automatique) | Entier | 2 |
| m\_localisationImages | Le tableau dynamique contenant les ImagesDansDiaporama | Vector <ImageDansDiaporama> | / |
| m\_posImage | La position de l’image courante | Entier | 0 |

|  |  |
| --- | --- |
| Nom méthode | Signification / but |
| Diaporama(string) | Constructeur par défaut. |
| Diaporama(const Diaporama&) | Constructeur par copie. |
| ~Diaporama() | Destructeur. |
| string getTitre() const | Getter permettant d’avoir le titre du diaporama. |
| vector<ImageDansDiaporama> getLocalisationImages() | Getter permettant d’avoir la localisation des images. |
| unsigned int getPosImageCouranteInt() | Getter permettant d’avoir le numéro de la position de l’image courante. |
| unsigned short int getVitesseDefilement() | Getter permettant d’avoir la vitesse de défilement d’une image. |
| ImageDansDiaporama getPositionImage() const | Getter permettant d’avoir la position d’une image. |
| unsigned int getNombreImages() const | Getter permettant d’avoir la taille du vecteur des images. |
| void setTitre(const string &) | Setter pour définir le titre du diaporama à partir de la chaine de caractère en paramètre. |
| void setVitesseDefilement(unsigned int) | Setter pour définir la vitesse de défilement d’un diaporama. |
| void setLocalisationImages(const vector<ImageDansDiaporama> &) | Setter pour définir un vecteur d’images. |
| void setPosImageCouranteInt(const unsigned int&) | Setter pour définir le numéro de la position de l’image courante. |
| void ajouterImage(const ImageDansDiaporama&) | Ajouter une image dans un vecteur d’images. |
| void afficherImageCouranteDansDiaporamaCourant() | Affiche l’image courante du diaporama courant. |
| void avancer() | Permet d’avancer dans le diaporama. |
| void reculer() | Permet de reculer dans le diaporama. |
| void triCroissantRang() | Permet de trier les images de manière croissante dans le vecteur d’images. |
| unsigned int nbImages() const | Permet d’avoir le nombre d’images dans le diaporama. |

# Version 2 - projet version graphique interface complète

## 2.1. Diagramme état-transitions classique

Une image contenant diagramme, texte, ligne, Parallèle

Description générée automatiquement

## Diagramme état-transitions matriciel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Éléments d’interface déclencheurs des événements | actionCharger\_diaporama | | actionEnlever\_le\_diaporama | rbModeDefilement | actionQuitter |
| Evénement  Etat | chargerDiaporama | changerDiaporama | viderDiaporama | Changer Mode | Quitter |
| diaporama non chargé | diaporama chargé | --- | --- | --- | Quittée |
| diaporama chargé | --- | diaporama chargé | diaporama non chargé | Défilement manuel | --- |
| Défilement automatique | --- | --- | --- | Défilement manuel | Quittée |
| Défilement manuel | --- | --- | --- | Défilement automatique | Quittée |
| Quittée | --- | --- | --- | --- | --- |

## Liens entre éléments d’interface et fonctionnalités

Utilité des boutons :

« Précédent » : Permet de retourner à l’image précédente lorsque l’on est en mode de défilement manuel.

« Pause/Play » : Permet de mettre le défilement en pause lorsque l’on est en mode de défilement automatique.

« Suivant » : Permet de passer à l’image suivante lorsque l’on est en mode de défilement manuel.

Utilité des radio-boutons :

« Défilement automatique » : Permet de changer de mode de défilement pas défaut le boutons et décoché (non activé) ce qui veut dire que l’on est en mode manuel.

« Boucle » : Permet d’indiquer si l’on souhaite retourner à la première image lorsque l’on a atteint la dernière image du diaporama.

Utilité des actions de la barre de menu :

Fichier > « Quitter » : permet de fermer l’application.

Paramètres > « Charger diaporama » : Permet de charger un nouveau diaporama (qu’il y en ait un déjà chargé ou pas).

Paramètres > « Vider diaporama » : Permet d’enlever le diaporama qui est déjà chargé.

Paramètres > « Vitesse de défilement » : Permet d’ouvrir une boite de dialogue qui donne la possibilité de changer la vitesse de défilement des images du mode de défilement automatique.

Aide > « A propos » : Permet d’afficher la version actuelle du lecteur de diaporama et ses auteurs.

# Version 2 - modèle MVP

### 3.1. Diagramme des classes UML

## Attributs et méthodes des classes