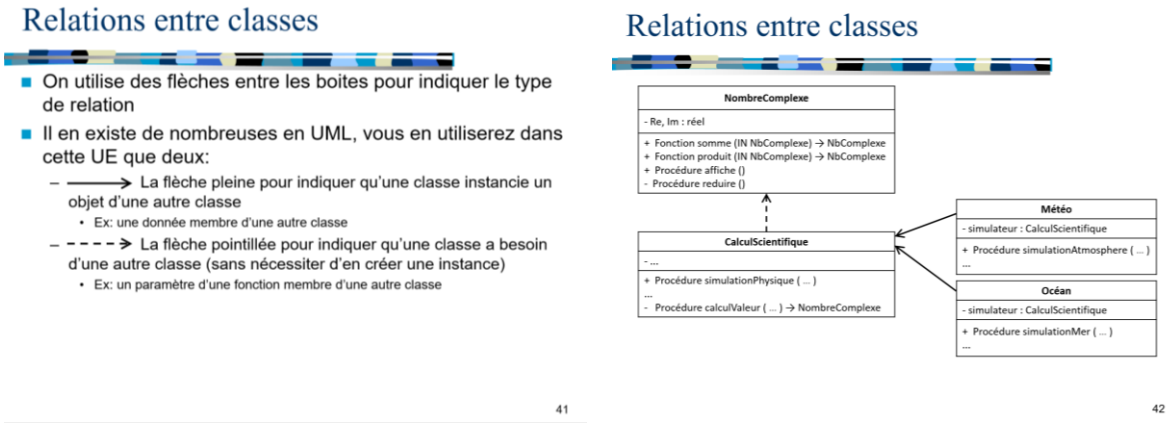
TD Conception



Rappel de cours sur les relations entre classes

Exercice 1

On désire stocker un ensemble de voitures dans un conteneur de type tableau dynamique. Une voiture est caractérisée par une marque, un modèle, une date de mise en circulation, un kilométrage, une couleur, l'ensemble des conducteurs présents et passés. Les fonctionnalités désirées sont : ajouter une voiture au conteneur, rechercher la plus veille voiture, celle avec le plus de km, et calculer la moyenne d'âge des voitures. Ecrivez le diagramme des classes correspondant.

Exercice 2

Une médiathèque vous demande de développer son logiciel de gestion des emprunts. Les employés de la médiathèque vous expliquent qu'ils aimeraient pouvoir gérer :

- Une liste d'emprunteurs avec leur nom, prénom, adresse (pour les retrouver en cas de problème) et leur âge (à des fins de statistique)
- Une liste de média avec le titre, l'auteur et le type : CD, livres, cassettes vidéo ou DVD
- Un historique des emprunts : la personne qui emprunte, le média emprunté, la date de sortie et la date de retour (ou la date de retour prévue si le média est encore sorti)

Vous êtes plusieurs développeurs sur ce projet, en tant que chef de projet vous devez spécifier au mieux les différentes classes nécessaires et leurs interactions afin que chaque développeur puisse commencer à travailler. Ils ont besoin du graphe de dépendance et de la spécification de toutes les classes (données et fonctions membres publiques et privées).

Exercice 3

La scolarité de la Faculté des Sciences et Technologies de Lyon 1 souhaite informatiser la gestion des étudiants, des enseignants et des UEs. Vous êtes chargés de développer une application en C++. Il faut notamment pouvoir représenter :

- L'ensemble des UEs proposées (code, intitulé, contenu, n° séquence, nombre de crédits ECTS, nombre de CM/TD/TP)
- L'ensemble des enseignants intervenants (état civil, n° employé, statut)

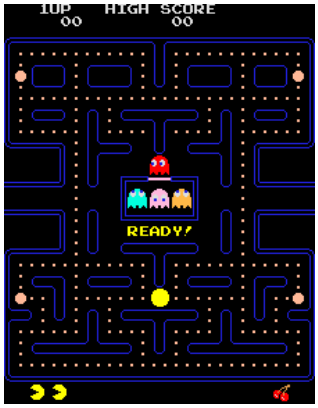
- La liste des étudiants inscrits dans l'UFR (état civil, n° étudiant)
- Quel étudiant est (ou a été inscrit) à telle UE pendant tel semestre. On souhaite également faire un suivi des étudiants (notes, absences, remarques éventuelles)
- Quel enseignant intervient (ou est intervenu) dans telle UE pendant tel semestre

Ecrivez le diagramme des classes, en détaillant les données membres. Faîtes apparaître les dépendances entre classes.

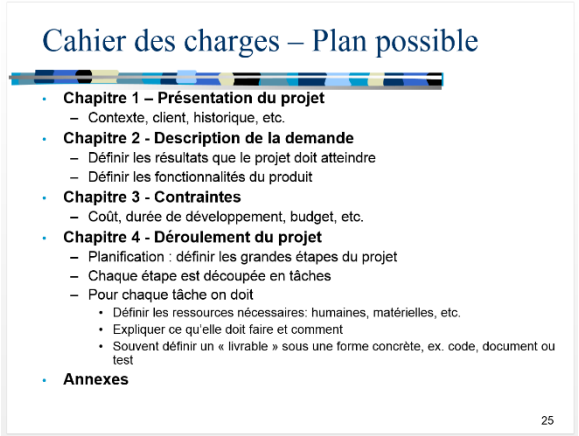
Exercice 4

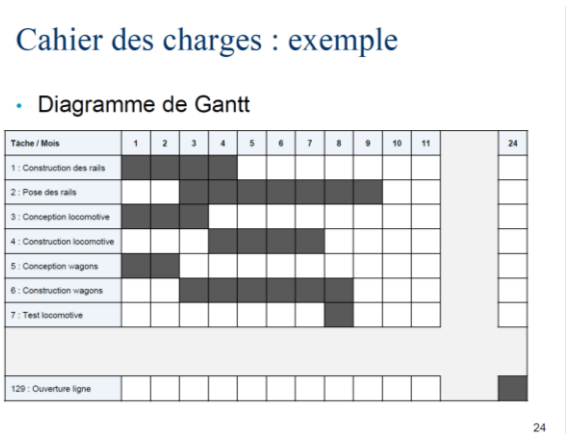
Vous êtes chef du projet **Pacman**. Pour le cahier des charges :

- Faire le diagramme des classes
- Découper le projet en tâches et faire le diagramme de Gantt

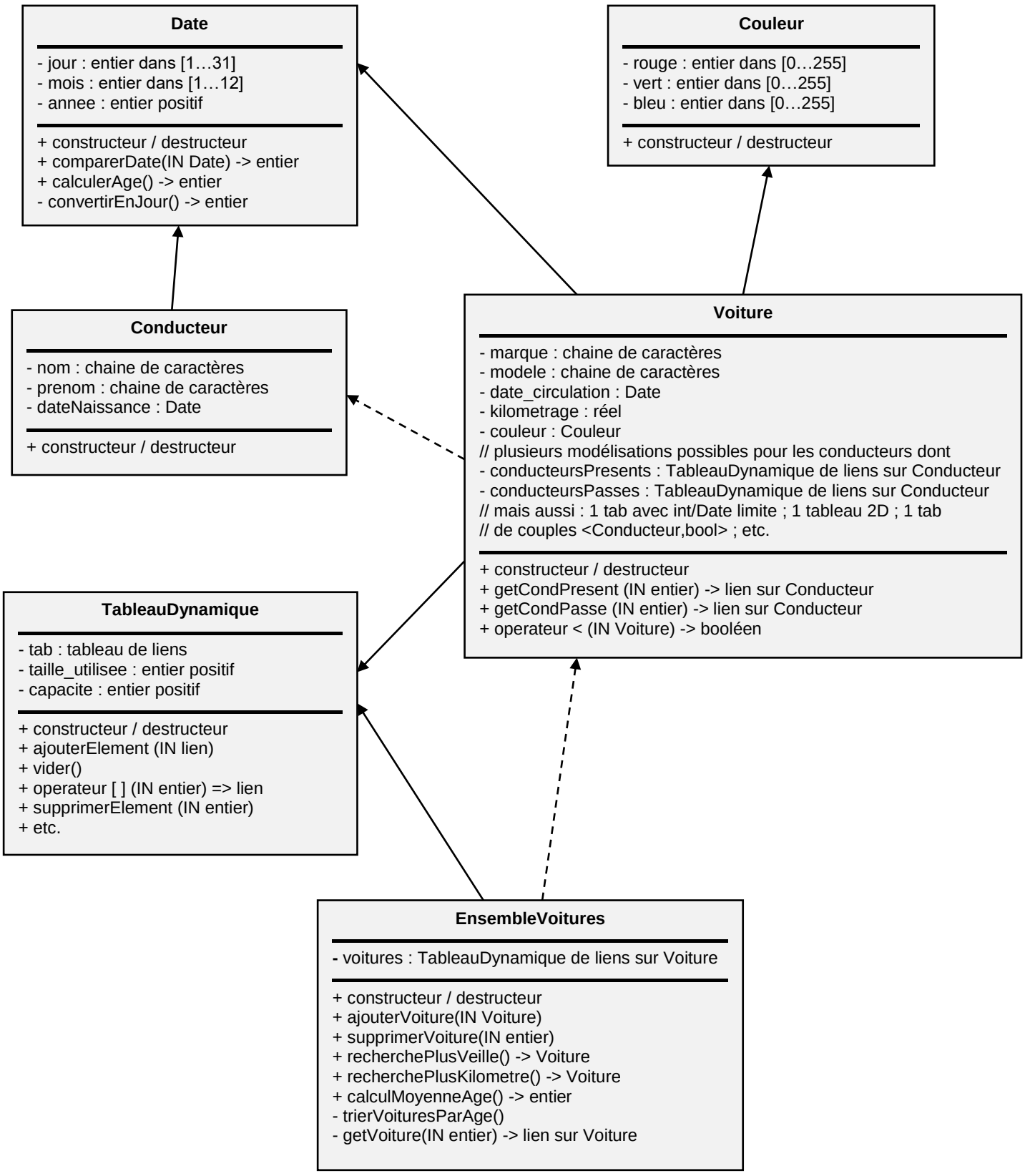


Rappel de cours :
- un plan possible de cahier des charges (à gauche)
- le chapitre 4 contient le diagramme de Gantt (exemple à droite) et le diagramme des classes





Exercice 1 – Voitures, diagramme des classes



Exercice 2 – Médiathèque
Liste des classes

Date

- jour : entier dans [1…31]
- mois : entier dans [1…12]
- annee : entier positif

+ constructeur / destructeur
+ getChaine () -> chaine de caractères

Media

- num_media : entier
- titre : chaine de caractères
- auteur : chaine de caractères
- type : entier dans {LIVRE, CD, DVD, VHS}

+ constructeur / destructeur
+ getNumMedia() -> entier
+ getTitre () -> chaine de caractères
+ getAuteur () -> chaine de caractères
+ getType () -> entier

Personne

- num_personne : entier
- nom : chaine de caractères
- prenom : chaine de caractères
- adresse : chaine de caractères
- dateNaissance : Date

+ constructeur / destructeur
+ getNumPersonne() -> entier
+ getNom () -> chaine de caractères
+ getPrenom () -> chaine de caractères
+ getAdresse () -> chaine de caractères
+ setAdresse (IN chaine de caractères)

Emprunt

- num_emprunt : entier
- num_media : entier
- num_personne : entier
- date_debut : Date
- date_fin : Date

+ constructeur / destructeur
+ getNumEmprunt () -> entier
+ getNumMedia () -> entier
+ getNumPersonne () -> entier
+ getDateDebut () -> Date
+ getDateFin () -> Date
+ setDateFin (IN Date)

ListeMedias

- tab_Medias : tableau de Media
- nb_Medias : entier

+ viderListeMedias()
+ getNbMedias() -> entier
+ getMedia (IN entier) -> Media
+ ajouterMedia (IN Media)
+ supprimerMedia(IN entier)

ListePersonnes

- tab_Personnes : tableau de Personne
- nb_Personnes : entier

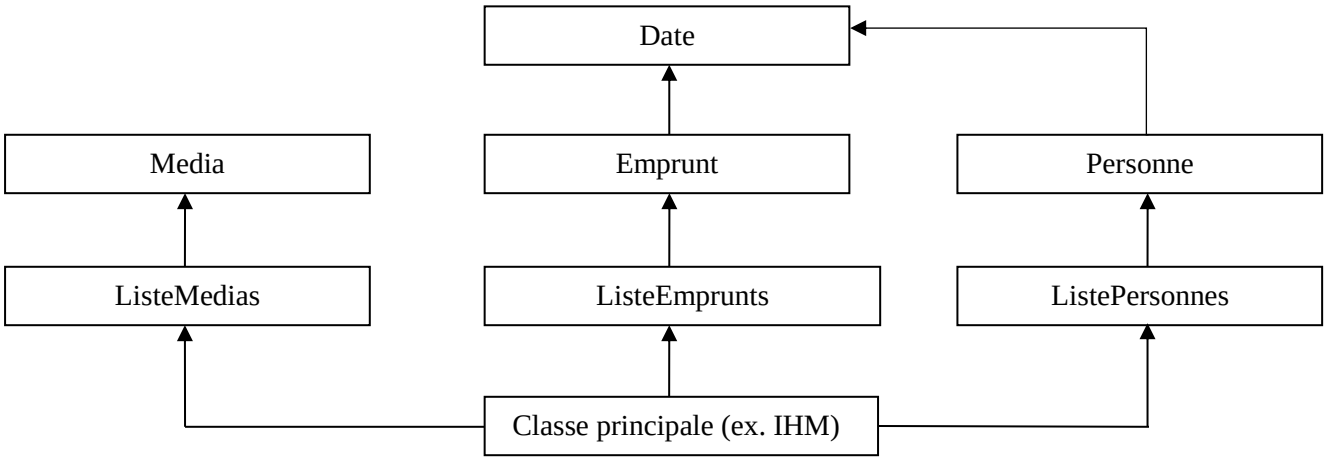
+ viderListePersonnes()
+ getNbPersonnes() -> entier
+ getPersonne(IN entier) -> Personne
+ ajouterPersonne(IN Personne)
+ supprimerPersonne(IN entier)

ListeEmprunts

- tab_emprunts : tableau d'Emprunt
- nb_emprunts : entier

+ viderListeEmprunts()
+ getNbEmprunts() -> entier
+ getEmprunt(IN entier) -> Emprunt
+ ajouterEmprunt(IN Emprunt)
+ supprimerEmprunt (IN entier)

Diagramme des dépendances



Exercice 4 – Pacman, éléments de réponse pour le cahier des charges

1. Présentation des acteurs

La société InfogamesProf est une Société Anonyme au capital de 1'000'000 d'euros. Son activité principale est l'édition de jeu vidéo sur PC. blahblah...

La société UnivGamesEtu est une start-up issue de l'université Claude Bernard Lyon 1. Elle est composée de 4 développeurs à plein temps. Son activité principale est le développement de produits informatiques pour des grands groupes. Son expertise est essentiellement centrée sur le C/C++ et le développement de listes chaînées et de tableau dynamique. blahblah ...

Dans ce contexte, la société InfogamesProf commande le produit « Pacman en folie » à la société UnivGamesEtu. Le présent cahier des charges est composé :

- d'une description détaillée du produit
- d'une prévision détaillée du déroulement du développement
- d'un diagramme de Gantt
- d'un diagramme des classes

L'ensemble de ces documents sera maintenu pendant toute la durée de développement du produit.

2. Description du produit

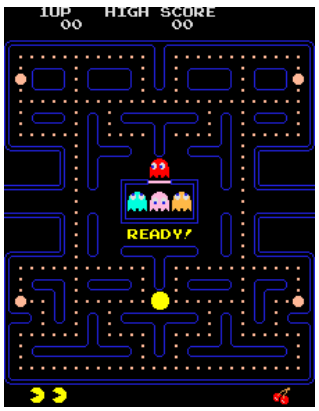
Principe et règles du jeu

Le jeu commence en cliquant sur l'écran de jeu. Le but du jeu est de collecter tous les points dans chaque niveau. Pendant ce temps on est suivi par plusieurs fantômes qui ne doivent pas nous toucher.

Le Pacman est déplacé grâce aux touches « flèches ». Partout où il passe, il mange les pastilles jaunes. Lorsque toutes les pastilles ont été mangées, le niveau est terminé et on passe au suivant. Pacman se transforme en super-Pacman en mangeant les pastilles blanches. Super-Pacman peut manger les fantômes. Super-Pacman redevient Pacman après 10 secondes. Les fantômes ont un déplacement automatique en direction de Pacman pour le manger ou alors en direction opposée à super-Pacman pour ne pas se faire manger. Le jeu se déroule avec une musique et des bruitages en fonction de qui mange qui. Le jeu se termine lorsque l'on a perdu ses trois vies.

Détail de l'interface et des niveaux du jeu

Ci-contre un exemple du niveau 1



Croquis des niveaux (au stylo par exemple) ...

Description de la gestion des scores ...
Description des autres fonctionnalités ...

3. Contraintes

- Le jeu sera développé en C/C++ sous Linux
- Les librairies utilisées seront SDL2 et une librairie texte pour une version alpha
- Le code respectera le standard suivant : code indenté, variable ayant du sens, ...
- Le code sera géré et archivé sur GitLab
- La documentation du code sera produite par Doxygen
- Un diagramme des classes permettra d'avoir une vision de haut niveau de l'implantation
- L'équipe de développement utilisera les outils de debug et de profiling : gdb et valgrind
- Le code sera fourni à la société InfogamesProf

4. Déroulement du projet

Tâche 0 : rédiger le cahier des charges

Membres impliqués : tous
Durée : 2 semaines

Tâche 1 : définir le diagramme des classes

Membres impliqués : tous
Durée : 1 semaine

Tâche 2 : développement d'un 1er prototype en mode texte

Durée : 3 semaines

Tâche 2.1 : écriture et test du module Pacman (`Pacman.h/.cpp`)
Membres impliqués : Bob Lecodeur
Le Pacman est défini par une position, un état, un score, blahblah
Le Pacman peut se déplacer vers la gauche, la droite blahblah

Tâche 2.2 : écriture et test du module Terrain (`Terrain.h/.cpp`)
Membres impliqués : Pierre Lerouleur et Joe Lefort
Ce module pourra charger un terrain depuis un fichier. Le format d'un fichier de données respectera la syntaxe suivante : blahblah

Tâche 2.3 : écriture et test du module Fantôme (`Fantome.h/.cpp`)
Membres impliqués : Bob Lecodeur et Pierre Lerouleur
Définition des classes pour un Fantôme: blahblah

Tâche 2.4 : écriture et test du module Jeu (`Jeu.h/.cpp`)
Membres impliqués : blahblah
Ce module propose toutes les fonctions pour gérer une étape du jeu :
- reçoit en entrée ce qu'un utilisateur a tapé au clavier
  => une structure sera définie à cet effet
- déplacement de Pacman en utilisant le module Pacman et Terrain
- déplacement d'un fantôme en utilisant le module Fantôme et Terrain
- mise à jour du score
- test de fin de niveau

Attention : ce module ne fait aucun affichage. Il doit pouvoir servir à la version texte comme à la version graphique.

Tâche 2.5 : écriture du module JeuModeTexte (`JeuModeTexte.h/.cpp`)
Membres impliqués : blahblah
Récupère les données de Terrain, Pacman et Fantôme et les affiche au format texte.
blahblah

Tâche 2.6 : mise en commun des modules et tests. blahblah

Tâche 3 : développement d'un 2e prototype en mode graphique (SDL2)

Durée : 4 semaines

Tâche 3.1 : exploration et compréhension de SDL2
blahblah

Tâche 3.2 : Affichage d'images et tests de performance
blahblah

Tâche 3.3 : Développement d'une boucle de jeu en SDL2 dans le module JeuModeGraphique (`JeuModeGraphique.h/.cpp`) et intégration au reste du jeu
blahblah

Tâche 3.4 : Test et debug
blahblah

Tâche 4 : développement d'un 3e prototype en mode graphique avec la musique et gestion des scores

blahblah

Tâche 5 : développement d'un 4e prototype avec ajout d'un mode multi-joueurs

blahblah

Tâche 6 : développement d'un 5e prototype multi-joueurs en réseau

blahblah

5. Diagramme de Gantt

	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7
Tâche 0	X	X					
Tâche 1		X					
Tâche 2.1		X	X				
Tâche 2.2		X	X				
Tâche 2.3		X	X				
Tâche 2.4			X	X			
Tâche 2.5			X	X			
Tâche 3.1		X	X	X			
etc.							

6. Diagramme des classes (version simplifiée)