TP4 - Héritage et Polymorphisme
ING2 - GSI - 2023-2024

1 Puissance Electrique

Définir une classe de base `Electrique` comportant des données `Tension` et `Intensite` l'utiliser pour les pour les classes dérivées `Ampoule` et `Guirlande`. La `Tension` par défaut est de `220V` en courant continu, les intensités sont variables et mesurées sur notre première `Guirlande,` les mesures de nos deux premières `Ampoule` sont `0,05A` et `0,1A`.

Sachant qu'une ampoule éteinte ne consomme pas d'électricité, utiliser une fonction virtuelle de la classe `Electrique` pour calculer la `Puissance` consommé pour une `Ampoule`.

Reprendre le scénario de test implanté dans la fonction main du TP3 :

```
.o
o.
o.o.o.o.o.
.o.o.o.o.o
...
.o.
.o..o..o..o..o.
.o..o..o..o.
```

Changer la représentation des `Ampoule` en utilisant le symbole . pour les ampoules éteintes, `o` pour les ampoules ayant une intensité <= à `0,05A` et le symbole `O` les ampoules > à `0,1A`, ajouter une fonction pour allumer toutes les ampoules.

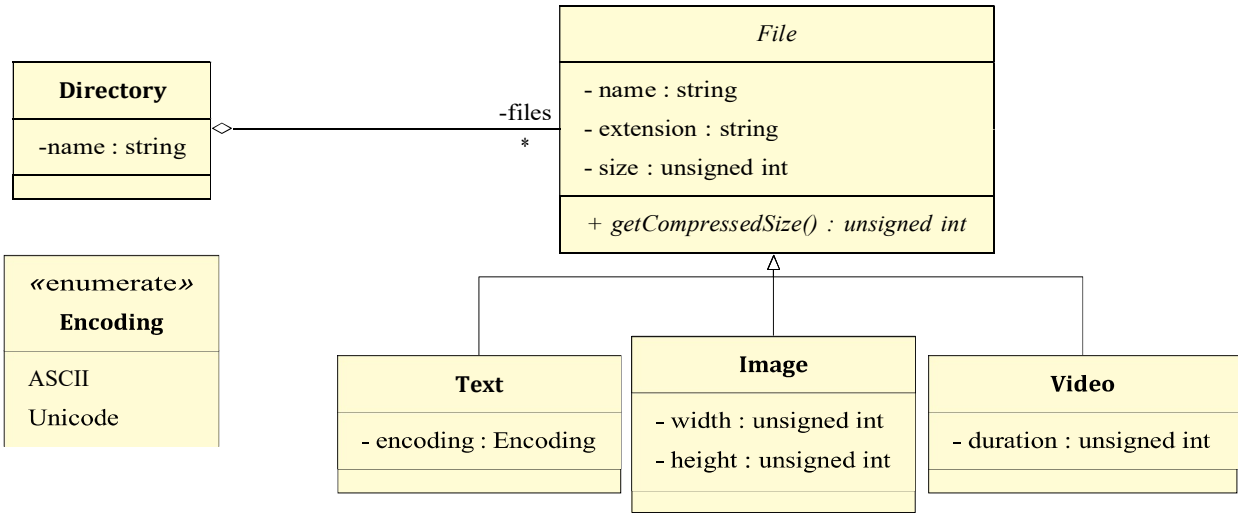
Résultat attendu :

```
.O
o.
o.o.o.o.o.
.O.O.O.O.O
...
.O.
.O..O..O..O..O.
.O..O..O..O.
oOooOooOooOo
```

En fonction de la puissance électrique de chaque `Ampoule` déduire et afficher la puissance consommée pour chaque `Guirlande`.

2 Classes des fichiers

Nous souhaitons modéliser les répertoires et les différents types de fichier dans un système d'exploitation, avec le diagramme de classes UML suivant. L'association entre un répertoire et ses fichiers est uneagrégation car nous voudrons gérer les liens symboliques également.



Nous inventons une fonction fictive *getCompressedSize()* qui retourne la taille du fichier après avoir appliqué un algorithme de compression et ce résultat dépend de type de fichier:

Texte : si l'encodage est de type ASCII, la taille sera de 50% de la taille initiale, sinon 60%;

Image : la taille du fichier compressé sera de la moitié du produit *width*height* ;

Vidéo : la taille du fichier compressé sera 80% de la durée de la vidéo.

1. Programmez ces classes en C++ en ajoutant des constructeurs, destructeurs et des méthodes nécessaires (*getter*, *setter*, affichage, ...) et en respectant les attributs et les méthodes données. □

2. Créez des jeux de test dans une fonction main.cpp pour ajouter des fichiers dans un répertoire principal, puis :

   a. Calculer le taux de compression moyen dans ce répertoire (le taux est calculé par la formule *getCompressedSize()/size*).

   b. Calculer le taux moyen de compression des fichiers vidéos dans ce dossier. □