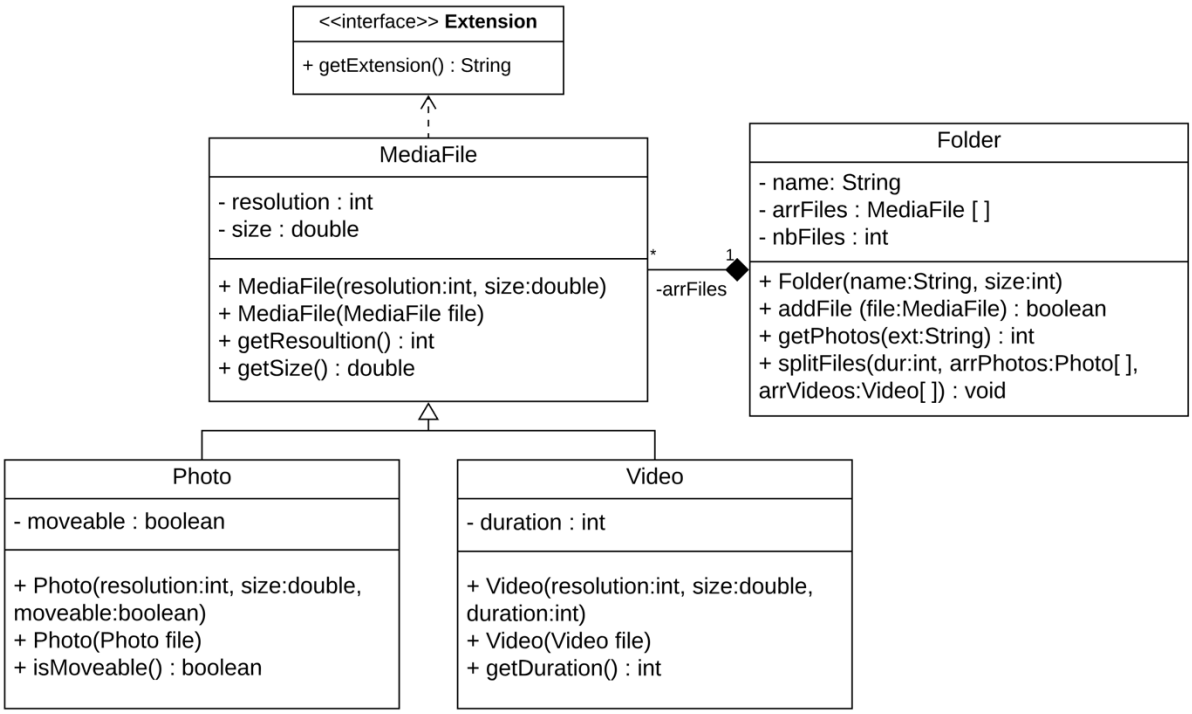

Q1: Write the interface "Extension", and the classes "MediaFile", "Photo", and "Video" in java. The classes descriptions are as follows (**5 marks**):

- Interface *Extension*
  - Methods:
    - getExtension() : this methods returns the extension of the file as follows:
      - For a **Photo:**
        - If moveable = true, it returns ".gif".
        - If moveable = false, it returns ".jpg".
      - For a **Video:**
        - If size is larger than 2000, it returns ".avi".
        - If size is less than or equals to 2000, it returns ".mp4".
- Class *MediaFile:*
  - Attributes:
    - resolution: stores the resolution of the media file as an int value.
    - size: stores the size of the media file as a double value.
  - Methods:
    - MediaFile(resolution:int, size:double) : constructor.
    - MediaFile (MediaFile file) : copy constructor.
    - getResolution() : return the resolution of the media file, but if the resolution is negative, it should throw **IllegalArgumentException**.
    - getSize() : returns the size of the media file.

- Class *Photo*:
  - Attributes:
    - moveable: stores true if the photo is moveable and false otherwise.
  - Methods:
    - Photo(resolution:int, size:double, moveable:boolean): constructor.
    - Photo (Photo file) : copy constructor.
    - isMoveable(): returns the value of moveable.

- Class *Video*:
  - Attributes:
    - duration: stores the duration of the video as an int value.
  - Methods:
    - Video(resolution:int, size:double, duration:int) : constructor.
    - Video (Video file) : copy constructor.
    - getDuration() : returns the duration of the video.

Q2: Write the class "Folder" in java. The class description is as follows (**5 marks**):

- Class *Folder*:
  - Attributes:
    - name: stores the name of the folder.
    - arrFiles: array of media files.
    - nbFiles: number of media files saved in this folder.
  - Methods:
    - Folder(name:String, size:int) : constructor.
    - addFile(file:MediaFile) : add a new media file to the folder. This method returns true if the file is added, and **throw IndexOutOfBoundsException** if the folder is full.
    - getPhotos(ext:String) : return the number of photos that have the same extension as **ext**.
    - splitFiles(dur:int, arrPhotos:Photo[ ], arrVideos:Video[ ]) : this method stores all the photos in the array 'arrPhotos', and stores all the videos that have a duration less than **dur** in the array 'arrVideos'.

When you finish, export your project as a '.zip' file (File > export > general > archive file) and submit it to the blackboard website (https://lms.ksu.edu.sa/).