HW:

- Create abstract class `MediaContent`
  - methods: play(), getDuration()
  - attributes: title, releaseYear, duration
- Create interface: `Downloadable`
  - methods: download()
- Class `Movie` (extends MediaContent, implements Downloadable)
  - attributes: Genre, Director
  - bonus: can you limit the number of acceptable Genres to Comedy and Action?
- class `TVSeries` (extends MediaContent, implements Downloadable)
  - attributes: seasons, episodesPerSeason
- class `Documentary` (extends MediaContent)
  - attributes: category, narrator
- Main class:
  - Create an array of `MediaContent` with items from all 3 classes
  - Loop through and call play() and getDuration() for each object
  - Use instanceof to check for `Downloadable` objects and if available, download it

```
if (media instanceof Downloadable) {
    ((Downloadable) media).download();
}
```

HW 2:

- Create a Zoo class with:
  - Abstract Class Animal
    - methods `makeSound`, `getDiet`
  - Interface Trainable
    - method `performTrick`
  - Class Mammal (extends Animal)
    - Attribute: furType
    - Implements methods from Animal with “generic mammal sound” and “omnivore”
  - Class Bird (extends Animal, implements Trainable)
    - attribute: wingspan (double)
    - getDiet, performTrick return “Insectivore” and “flying in circles!”
  - Class Lion (extends Mammal)
    - makeSound, getDiet returns “Roar” and “Carnivore”
  - Class Parrot (extends Bird)
    - makeSound, getDiet return “Squawk” and “Herbivore”
    - performTrick returns “the Parrot is mimicking sounds!”
  - Main:
    - Create an Array of Lion, Parrot, Mammal, Bird
    - Loop through and print the results of calling `makeSound`, `getDiet`
    - Use `instanceOf` to call `performTrick` on Trainable objects