# P2W1 Lambda/Stream – Opgave “Acteur”

We wensen een aantal *manipulaties* te doen op een **`ArrayList`** met objecten van de klasse **`Acteur`**.

Zoals je hieronder kunt zien is de klasse **`Acteur`** is reeds voorzien van de nodige methoden om de opdracht tot een goed einde te brengen (**`equals`**, **`hashCode`** en **`compareTo`**).

## De Klasse Acteur

```
public class Acteur implements Comparable<Acteur> {
    private final String naam;
    private final int geboorteJaar;

    public Acteur(String naam, int geboorteJaar) {
        this.naam = naam;
        this.geboorteJaar = geboorteJaar;
    }

    public int getGeboorteJaar() {
        return geboorteJaar;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Acteur acteur = (Acteur) o;
        return geboorteJaar == acteur.geboorteJaar &&
                Objects.equals(naam, acteur.naam);
    }

    @Override
    public int hashCode() {
        return Objects.hash(naam, geboorteJaar);
    }

    public int compareTo(Acteur andere) {
        int jaarVerschil =
                        Integer.compare(geboorteJaar, andere.geboorteJaar);
        if (jaarVerschil != 0) return jaarVerschil;
        return naam.compareTo(andere.naam);
    }

    @Override
    public String toString() {
        return geboorteJaar + "  " + naam;
    }
}
```

Vul het gevraagde aan in de klasse **TestActeur**. Voor het **Stream** –gedeelte dien je als volgt tewerk te gaan:

- Gebruik de methode **filter** uit de interface **Stream<T>** , de signatuur hiervan is **Stream<T> filter(Predicate<? Super T> predicate)**
- Voor de volgende twee stappen zal je gebruik moeten maken van andere Stream<T> methoden, zoek op in de Javadoc of…
- Als laatste stap dien je de stream om te zetten naar een ArrayList. Hiervoor maak je gebruik van de collect methode uit de interface **Stream<T>**, de signatuur hiervan is **<R,A> R collect(Collector<? Super T,A,R> collector)**
- Als parameter van deze laatste methode gebruik je een methode uit de klasse **Collectors**. Met behulp van deze methode zal je alle elementen van de stream in een nieuwe **ArrayList** accumuleren en die **ArrayList** uiteindelijk aan de variabele **acteurs** toekennen.

## De klasse TestActeur

```
public static void main(String[] args) {
        Acteur reese = new Acteur("Reese Witherspoon", 1976);
        Acteur drew = new Acteur("Drew Barrymore", 1975);
        Acteur anna = new Acteur("Anna Faris", 1976);
        Acteur thandie = new Acteur("Thandie Newton", 1972);

        List<Acteur> acteurs = new ArrayList<>();

        acteurs.addAll(Arrays.asList(testdata));
        acteurs.add(reese);
        acteurs.add(drew);
        acteurs.add(anna);
        acteurs.add(thandie);

        // Toon de inhoud van de collection (gebruik forEach zonder Stream)


        // Verwijder de objecten reese en thandie (reeds uitgewerkt)
        acteurs.remove(reese);
        acteurs.remove(thandie);

        // Verwijder alle acteurs geboren in 1975
        // Verwijder alle acteurs die meer dan éénmaal voorkomen
        // Sorteer volgens geboortejaar (oudste eerst)
        // en dan volgens voornaam
        acteurs = acteurs.stream()


        // Toon de inhoud van de collection (gebruik forEach zonder Stream)


    }
}
```

Verwachte afdruk:

```
1972  Cameron Diaz
1976  Anna Faris
1975  Angelina Jolie
1970  Jennifer Lopez
1976  Reese Witherspoon
1973  Neve Campbell
1969  Catherine Zeta-Jones
1982  Kirsten Dunst
1975  Kate Winslet
1975  Gina Philips
1973  Shannon Elisabeth
1972  Carmen Electra
1975  Drew Barrymore
1965  Elisabeth Hurley
1975  Tara Reid
1978  Katie Holmes
1976  Anna Faris
1976  Reese Witherspoon
1975  Drew Barrymore
1976  Anna Faris
1972  Thandie Newton
```

Uiteindelijke inhoud:

```
1965  Elisabeth Hurley
1969  Catherine Zeta-Jones
1970  Jennifer Lopez
1972  Cameron Diaz
1972  Carmen Electra
1973  Neve Campbell
1973  Shannon Elisabeth
1976  Anna Faris
1976  Reese Witherspoon
1978  Katie Holmes
1982  Kirsten Dunst
```