



PMR: Once again this assignment was super easy since it was basically the previous two assignments. I just changed some of the movies to add more variety and some of the sorting methods. I enjoyed this assignment.

/\*\*

\* Movie Interface

\*

\* @author Anika Jallipalli

\* @version 4/11/2020

\*/

public class Movie4

{

private String title;

private int year;

private String studio;

public Movie4(String title, int year, String studio)

{

this.title = title;

this.year = year;

this.studio = studio;

}

public String toString()

{

String string = title + ", " + year + ", " + studio;

return string;

}

public String getTitle()

{

return title;

}

public int getYear()

{

return year;

}

public String getStudio()

{

return studio;

}

public void setStudio(String a)

{

studio = a;

}

}

/\*\*

\* run the program

\*

\* @author Anika Jallipalli

\* @version 4/11/2020

\*/

import java.lang.\*;

public class TestMovie4

{

public static void main(String[] args)

{

Movie4 muppets = new Movie4("The Muppets Take Mahattan",2001,"Columbia Tristar");

Movie4 mulan = new Movie4("Mulan Special Edition",2004,"Disney");

Movie4 shrek = new Movie4("Shrek 2",2004,"Dreamworks");

Movie4 incredibles = new Movie4("The Incredibles",2004,"Pixar");

Movie4 nanny = new Movie4("Nanny McPhee",2006,"Universal");

Movie4 rabbit = new Movie4("The Curse of Were-Rabbit", 2006,"Aardman");

Movie4 iceage = new Movie4("Ice Age",2002,"20th Century Fox");

Movie4 lilo = new Movie4("Lilo & Stitch",2002,"Disney");

Movie4 robots = new Movie4("Robots",2005,"20th Century Fox");

Movie4 monsters = new Movie4("Monsters Inc.",2001,"Pixar");

Movie4[] movies = new Movie4[] {muppets, mulan, shrek, incredibles, nanny,

iceage, lilo, robots, monsters, rabbit};

Movie4[] newmovies = new Movie4[movies.length];

System.out.println("Before Sorting:");

printMovies(movies);

System.out.println();

System.out.println("Sorted by Title - ascending:");

sortTitles(movies);

printMovies(movies);

System.out.println();

System.out.println("Sorted by Year - descending:");

sortYears(movies,0,9);

printMovies(movies);

System.out.println();

System.out.println("Sorted by Studios:");

sortStudios(movies);

printMovies(movies);

}

public static void printMovies(Movie4[] a)

{

for(int i = 0; i < a.length; i++)

{

System.out.println(a[i].toString());

}

}

public static void sortTitles(Movie4[] a)

{

if (a.length >= 2)

{

Movie4[] left = new Movie4[a.length / 2];

Movie4[] right = new Movie4[a.length - a.length / 2];

for (int i = 0; i < left.length; i++)

{

left[i] = a[i];

}

for (int i = 0; i < right.length; i++)

{

right[i] = a[i + a.length / 2];

}

sortTitles(left);

sortTitles(right);

mergeTitles(a, left, right);

}

}

public static void sortYears(Movie4[] a, int low, int high)

{

if ( low == high )

{

return;

}

int mid = ( low + high ) / 2;

sortYears( a, low, mid );

sortYears( a, mid + 1, high);

mergeYears( a, low, mid, high);

}

public static void sortStudios(Movie4[] a)

{

if (a.length >= 2)

{

Movie4[] left = new Movie4[a.length / 2];

Movie4[] right = new Movie4[a.length - a.length / 2];

for (int i = 0; i < left.length; i++)

{

left[i] = a[i];

}

for (int i = 0; i < right.length; i++)

{

right[i] = a[i + a.length / 2];

}

sortStudios(left);

sortStudios(right);

mergeStudios(a, left, right);

}

}

public static void mergeYears( Movie4[] a, int low, int mid, int high )

{

Movie4[] temp = new Movie4[ high - low + 1 ];

int i = low, j = mid + 1, n = 0;

while ( i <= mid || j <= high )

{

if ( i > mid )

{

temp[ n ] = a[ j ];

j++;

}

else if ( j > high )

{

temp[ n ] = a[ i ];

i++;

}

else if ( a[ i ].getYear() > a[ j ].getYear() )

{

temp[ n ] = a[ i ];

i++;

}

else

{

temp[ n ] = a[ j ];

j++;

}

n++;

}

for ( int k = low ; k <= high ; k++ )

{

a[ k ] = temp[ k - low ];

}

}

public static void mergeTitles(Movie4[] a, Movie4[] left, Movie4[] right)

int c = 0;

int d = 0;

for (int i = 0; i < a.length; i++) {

if (d >= right.length || (c < left.length && (left[c].getTitle()).compareToIgnoreCase(right[d].getTitle()) < 0))

{

a[i] = left[c];

c++;

} else {

a[i] = right[d];

d++;

}

}

}

public static void mergeStudios(Movie4[] a, Movie4[] left, Movie4[] right)

{

int c = 0;

int d = 0;

for (int i = 0; i < a.length; i++) {

if (d >= right.length || (c < left.length && (left[c].getStudio()).compareToIgnoreCase(right[d].getStudio()) < 0))

{

a[i] = left[c];

c++;

} else {

a[i] = right[d];

d++;

}

}

}

}