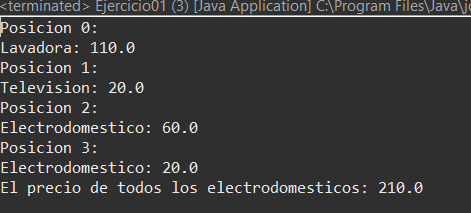
Tarea 09

Resumen del tema:

Creación objetos mediante clases y superclases otorgándoles atributos. Gestión de herencias.

Ejercicio 1:

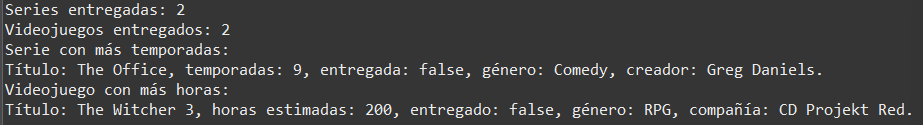
Creo la clase del ejercicio con el main, la superclass Electrodomestico, la class Lavadora y Television con versión extend para que dependan de la superclass y poder heredar sus atributos. Cada clase tiene sus atributos concretos, sus versiones por defecto, con 3 conductores. En el main creo una lista vacía con 4 posiciones, y luego llamo a 3 métodos. El primero rellena el array con posibles productos de forma random, el segundo muestra con un bucle el precio de cada producto y los muestra en la consola, y el tercero suma el precio de todos y lo muestra en la consola.



Ejercicio 2:

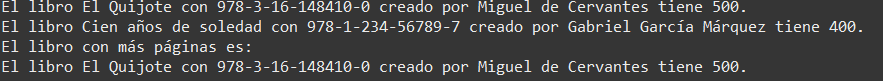
Creo una clase Serie y una Videojuego, con sus respectivos atributos, constructores, getters, setters y la sobrescripción del método toString.

Luego creo una interface con los métodos entregar, devolver, isEntregado, que es como el getEntregado, y compareTo. Los dos primeros sirven para volver true y false el atributo entregado, y el último compara las series y videojuegos según sus temporadas y horas para seleccionar las más longevas y mostrarlas en la consola. Creo en el main 2 arrays, uno para series y otro para videojuegos, con 5 posiciones cada uno, y les asigno los atributos que quiera. Y llamo a los métodos para ejecutar el código y mostrar las series y juegos entregados, la serie y el juego más longevo con todos sus dato,



Ejercicio 3:

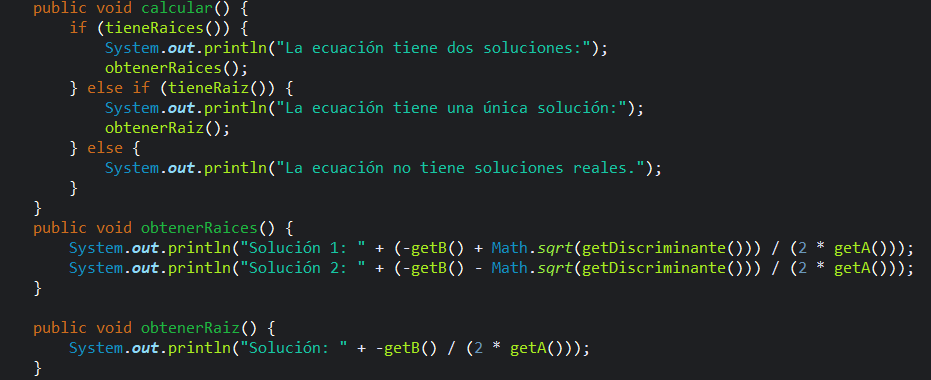
Creo una clase llamada Libro y le asigno los atributos correspondientes, getters y setters, hago un overide para modificar como se mostrará en la consola los datos de los libros y creo un constructor.  
Creo 2 objetos libro y les asigno los datos que quiero y creo una lista y meto los 2 objetos dentro. Luego creo un método que recorre el array y muestra los datos de los libros y luego llamo al método que compara los libros y dice cual de ellos tiene más páginas.



Ejercicio 4:

Creo una clase llamada Raices y le asigno 3 atributos double que son a, b y c. Y creo un constructor y los getters y setters correspondientes. Creo varios métodos, getDiscriminante que realiza una parte de la ecuación, tieneRaices, que devuelve un bool true si el discriminante es mayor que 0 de forma que la ecuación tiene 2 soluciones, tieneRaiz, que tiene el mismo funcionamiento, pero con la condición de que el discriminante sea igual a 0 por lo que la ecuación tiene solo una solución. También creo los métodos obtenerRaices y obtenerRaiz que muestran en la consola las soluciones posibles, siendo el primero cuando hay 2 y el segundo cuando hay 1.

Luego creo un super método llamado calcular donde llamo al resto y queda así:



Para acabar en el main creo un array que es igual al método asignarValores que he creado en el main para asignarle un valor double a la a, la b y la c, y devolviendo estos valores en un array. Llamo al constructor asignando a cada double el valor de la primera, segunda y tercera posición del array, creo un objeto para llamar al constructor y luego con ese objeto llamo al método calcular y todo se ejecuta correctamente.

Ejercicio 5:

El enunciado me pide organizar una clase de forma que haya un profesor y alumnos en un aula pero que estos pueden no asistir a clase por lo que puede no llevarse a cabo la clase y si hay clase mostrar los alumnos que han aprobado.  
Creo las clases Estudiantes, Profesores y Aula. Creo los atributos correspondientes, constructores y los getters y setters. Creo un método para la asistencia del profesor con un 20% de faltar y otro igual para los alumnos con un 50% de faltar. Hago 2 listas en el main con 10 alumnos y 3 profes con atributos a cada uno. Hago un método random que selecciona uno de los 3 profes y le aplica el método de si asistirá o no, otro método que recorre el array para mostrar los alumnos aprobados que asisten a clase y los separa en alumnos y alumnas.   
Todo esto lo junto en un supermétodo:

public void habraClase(Profesores[] profes, Estudiantes[] estudiantes) {

if (profes == null || profes.length == 0) {

System.out.println("No hay profesores disponibles, se suspende la clase.");

return;

}

Profesores profesorSeleccionado = seleccionarProfesor(profes);

if (profesorSeleccionado == null || !profesorSeleccionado.polimorfismoP()) {

System.out.println("El profesor no asistirá, se suspende la clase.");

} else {

System.out.println("El profesor asistirá a clase.");

if (estudiantes == null || estudiantes.length == 0) {

System.out.println("No hay estudiantes en la lista.");

return;

}

int contadorAsistentes = asistenciaEstudiantes(estudiantes);

if (contadorAsistentes < estudiantes.length / 2) {

System.out.println("No hay suficientes alumnos, se suspende la clase.");

} else {

System.out.println("La clase se llevará a cabo.");

listaAlumnos(estudiantes);

}

}

}

Con las condiciones de que si falta el profe no hay clase y no se ejecutan los métodos, y si no hay suficientes alumnos tampoco habrá clase.

Ejercicio 6: