Guía de Ejercicios LAB-121

Formato de entrega de informes

Universidad Mayor de San Andres
Facultad de Ciencias Puras y Naturales
Carrera Informatica

Logo de La UMSA

Practica N•

Nombre:
Materia:
Paralelo:
Docente:

La Paz - Bolivia

Para cada ejercicio realizar:

1. Enunciado
2. Diagrama de clases
3. Código*
4. Captura de pantalla

*Para las 2 primeras practicas realizar código manuscrito obligatoriamente

Laboratorio N° 1

1.- Sea la clase Fracción

```
/**
*       La Clase Fracción.
*/
public class Fraccion {
        private int  numerador;
        private int denominador;

        //Construye una nueva instancia de una Fraccion
        public Fraccion() {
                this.numerador = 3;
                this.denominador = 5;
        }

        //Retorna el numerador de la fracción
        public int getNumerador() {
                return numerador;
        }
        //Asigna el numerador de la fracción
        public void setNumerador(int numerador) {
                this.numerador = numerador;
        }


        //Retorna el denominador de la fracción
        public int getDenominador() {
                return denominador;
        }
        //Asigna el numerador de la fracción
```

```
public void setDenominador(int denominador) {
        this.denominador = denominador;
}
//Suma dos fracciones
public Fraccion sumar(Fraccion f) {
        Fraccion resultado = new Fraccion();
        resultado.setNumerador(this.numerador+f.getNumerador());
        resultado.setDenominador(this.denominador+f.getDenominador());
        return resultado;
}
//Resta dos fracciones
public Fraccion restar(Fraccion f) {
        Fraccion resultado = new Fraccion();
        resultado.setNumerador(this.numerador-f.getNumerador());
        resultado.setDenominador(this.denominador-f.getDenominador());
        return resultado;
}
//Multiplica dos fracciones
public Fraccion multiplicar(Fraccion f) {
        Fraccion resultado = new Fraccion();
        resultado.setNumerador(this.numerador*f.getNumerador());
        resultado.setDenominador(this.denominador*f.getDenominador());
        return resultado;
}



//Divide dos fracciones
public Fraccion dividir(Fraccion f) {
        Fraccion resultado = new Fraccion();
        resultado.setNumerador(this.numerador*f.getDenominador());
        resultado.setDenominador(this.denominador*f.getNumerador());
```

```
            return resultado;
        }
        //Retorna verdadero si el objeto o, es igual a esta Fracción
        public boolean igual(Object o) {
            if (!(o instanceof Fraccion)) {
                return false;
            } else {
                Fraccion f = (Fraccion) o ;
                return  f.numerador == this.numerador && f.denominador == this.denominador;
            }
        }
        //Retorna una cadena para poder imprimirla con los datos de la fraccion
        @Override
        public String toString() {
            return "Fraccion [numerador=" + numerador + ", denominador="
                    + denominador + "]";
        }
}
```

a) Adicione el siguiente método a la clase Fracción:

public double convertirADecimal()

Retorna una fracción convertida en un número decimal, por ejemplo si tenemos la fracción $\frac{1}{2}$ el método retornará 0.5, se debe validar fracciones divididas entre 0.

Adicione el siguiente método a la clase Fracción:

public Fraccion convertitAFraccion(double nro)

Retorna un número convertido en fracción, por ejemplo si tenemos un número 0.33 el método retornará $\frac{1}{3}$, se debe considerar números negativos.

Puesto que el tipo double del Java es impreciso, redondee el valor del número a una razonable cantidad de dígitos de precisión tal como 4; antes de convertirlos.

b) Adicione el siguiente método a la clase Fracción:

public boolean esInverso(Fraccion f)

Retorna si esta fracción es la inversa de la fracción f. Una fracción es inversa a otra si multiplicadas ambas el resultado de esta operación es 1. Un caso a considerar sería el de aquellas fracciones que tienen como denominador 0. En un caso más general puede ser determinado, calculando multiplicación de ambas fracciones y obteniendo 1 positivo cono resultado de dicha operación. Utilice el método multiplica(Fraccion f).

c) Adicione el siguiente método a la clase Fracción:

public static Fraccion parseFraccion(String str)

Convierta la cadena string a un apropiado objeto Fracción, el cual retornara de la función. Por ejemplo, Fraccion.parseFraccion("(-2/3)") debería retornar una nueva Fracción con el valor de *numerador* igual a -2 y el valor de *denominador* igual a 3. Esto debería devolver verdadero para cualquier Fracción

f. Fraccion.parseFracion(f.toString()).igual(f) .

d) Adicione el siguiente método a la clase Fracción:

public Fraccion simplifica()

El método deberá simplificar una Fracción y retornara una nueva función simplificada. Por ejemplo, si tenemos la fracción $\frac{2}{8}$ debería retornar una nueva Fracción con el valor igual a $\frac{1}{4}$. Para casos como $\frac{14}{28}$ se deberá simplificar hasta que ya no se pueda, debiendo retornar una nueva fracción $\frac{1}{2}$.

2.- Escriba la clase Figura2D que representa una figura geométrica en 2 dimensiones (rectángulo, cuadrado, triangulo y circulo). Los objetos Figura2D tienen los siguientes métodos:

public Figura2D(int x, int y, int val1, int val2, int val3 string tipo)

Construye una nueva Figura cuya coordenada de origen está especiada por los valores de (x,y), acompañado de los valores val1 y val2 que representaran su ancho y alto en caso de ser cuadrado o rectángulo; base y altura en caso de ser triangulo y centro para circulo y en cuyo caso se considerara el valor de val3 para el radio en los demás se pasara como nulo. Finalmente tipo nos dirá cuál de las 4 figuras geométricas se esta instanciando.

public int getX()

Retorna la coordenada x de la Figura2D.

public int getY()

Retorna la coordenada y de la Figura2D.

public int getVal1()

Retorna el val1 de la Figura2D.

public int getVal2()

Retorna el val2 de la Figura2D.

public int getVal3()

Retorna el val3 de la Figura2D.

public String toString()

Retorna una representación cadena de la Figura2D, tal como "Figura2D[Tipo=Rectangulo, x=1,y=2,ancho=3,alto=4]" , "Figura2D[Tipo=Circulo, x=5, y=3, centro=(3, 3), radio=6]"

O "Figura2D[Tipo=Triangulo, x=6, y=5, base=3, altura=7]".

a) Adicione el siguiente método a la clase Figura2D:

public boolean equals(Object o)

Retorna si el otro objeto o es una Figura2D igual a this.Figura2D (mismas coordenadas, mismos val1, val2, val3 y tipo).

b) Adicione el siguiente método a la clase Figura2D:

public Figura2D(Punto p, int[] vals, string tipo)

Construye una nueva Figura cuya coordenada de origen está especiada el Punto, acompañado de los valores del vector vals que representaran su ancho y alto en caso de ser cuadrado o rectángulo; base y altura en caso de ser triangulo y centro para circulo y en cuyo caso se considerara el valor de la tercera coordenada del vector para el radio. Finalmente tipo nos dirá cuál de las 4 figuras geométricas se está instanciando.

c) Adicione el siguiente método a la clase Figura2D:

public boolean contiene(int x, int y) public boolean contiene(Punto p)

Retorna si el Punto o la coordenada (x,y) está dentro de los límites de la Figura2D.

d) Adicione el siguiente método a la clase Figura2D:

public int Area()

Retorna el área de la Figura2D (Considere los 4 tipo de figuras ya que el área de un cuadrado no se la obtiene de la misma manera que la de un circulo).

e) Adicione el siguiente método a la clase Figura2D:

public int Perimetro()

Retorna el perímetro de la Figura2D (Considere los 4 tipo de figuras ya que el perímetro de un cuadrado no se la obtiene de la misma manera que la de un circulo).

3.- Definir las clases Docente, Estudiante, Auxiliar con sus respectivos atributos y métodos para cumplir:

a) Instanciar 2 objetos de cada clase con constructores diferentes.
b) Mostrar a los docentes, auxiliares y docentes que pertenecen a la misma carrera.
c) Mostrar a los estudiantes que son auxiliares.
d) Aumentar la materia dictada X al docente Y.
e) Mostrar al auxiliar del docente con nombre X.

Laboratorio N° 2

1. Sea la clase Concurso que tiene los siguientes atributos, como ser: fecha, nombre, tipo, nro de participantes, participantes, monto primer premio, monto segundo premio, monto tercer premio. Instanciar dos Concursos de manera diferente. Sobrecargar métodos para:

a) Incrementar una cantidad X de participantes al concurso.

b) Sobrecargar el método anterior para incrementar el monto del primer premio.

c) Sobrecargar el método anterior para incrementar la fecha del concurso.

2. Sea la clase Libro, identificar atributos necesarios y sobrecargar los métodos para:

a) Instanciar con 3 constructores diferentes 3 objetos Libros.

b) Mostrar el libro entero

b) Mostrar las k primeras páginas de un libro.

c) Mostrar las X primeras líneas de la página Y del libro.

d) Verificar si dos páginas son iguales.

e) Verificar si dos libros son iguales.

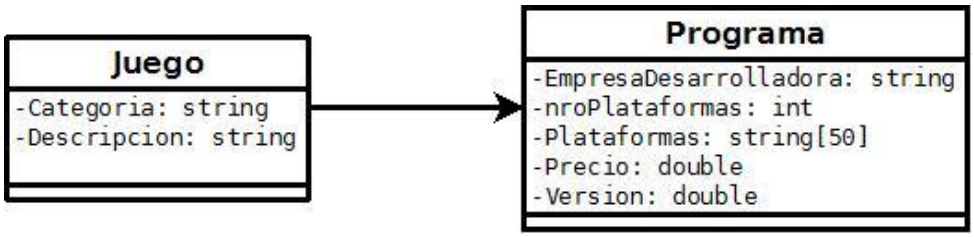
f) Verificar si existen líneas iguales de la página i.

3. Dada La clase biblioteca con nroLibros, Libros[2, 30](Del tipo String) y la clase Libro con nombre, autor, contenido.

a. Instanciar 1 biblioteca y 3 libros
b. Sobrecargar el operador ++ para leer por teclado (en cada clase)
c. Sobrecargar el operador -- para mostrar (en cada clase)
d. Sobrecargar el operador + para adicionar el libro A en la biblioteca
e. Sobrecargar el operador && para verificar si el libro B existe en la biblioteca

Laboratorio N° 3

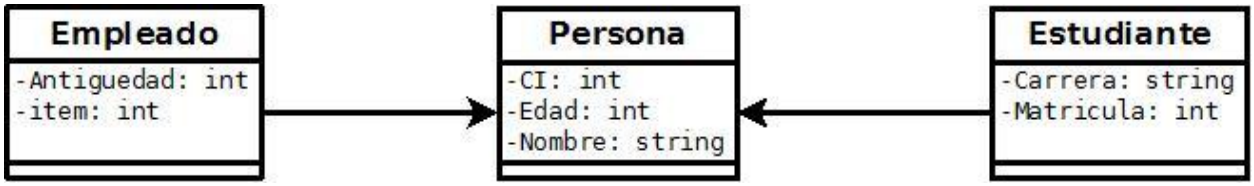
1. Sea el siguiente diagrama de clases:



Realizar métodos para:

a. Instanciar 2 Juegos de distinta manera.
b. Comparar 2 juegos y mostrar que juego está disponible en más plataformas.
c. Actualizar la versión de un Programa, actualizando sus atributos.
d. Dado un vector de Juegos, verificar y mostrar si existen juegos de la categoría X.

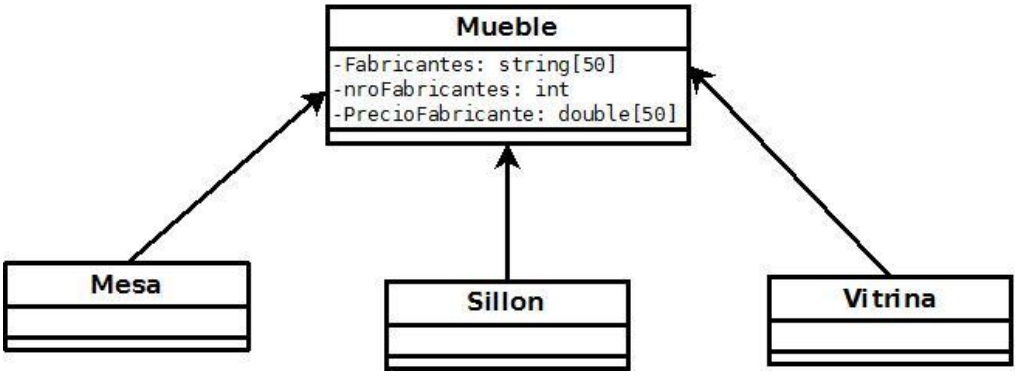
2. Sea el siguiente diagrama de clases:



Realizar métodos para:

a. Instanciar un vector de A Personas del tipo Empleado y B Personas del tipo Estudiante.
b. Mostrar al Empleado con mayor antigüedad.
c. Mostrar a todos los estudiantes ingresados en el Semestre X.
d. Eliminar del vector a los estudiantes de la carrera Z
e. Unir los 2 vectores en un mismo vector y ordenarlo por edad ascendentemente.

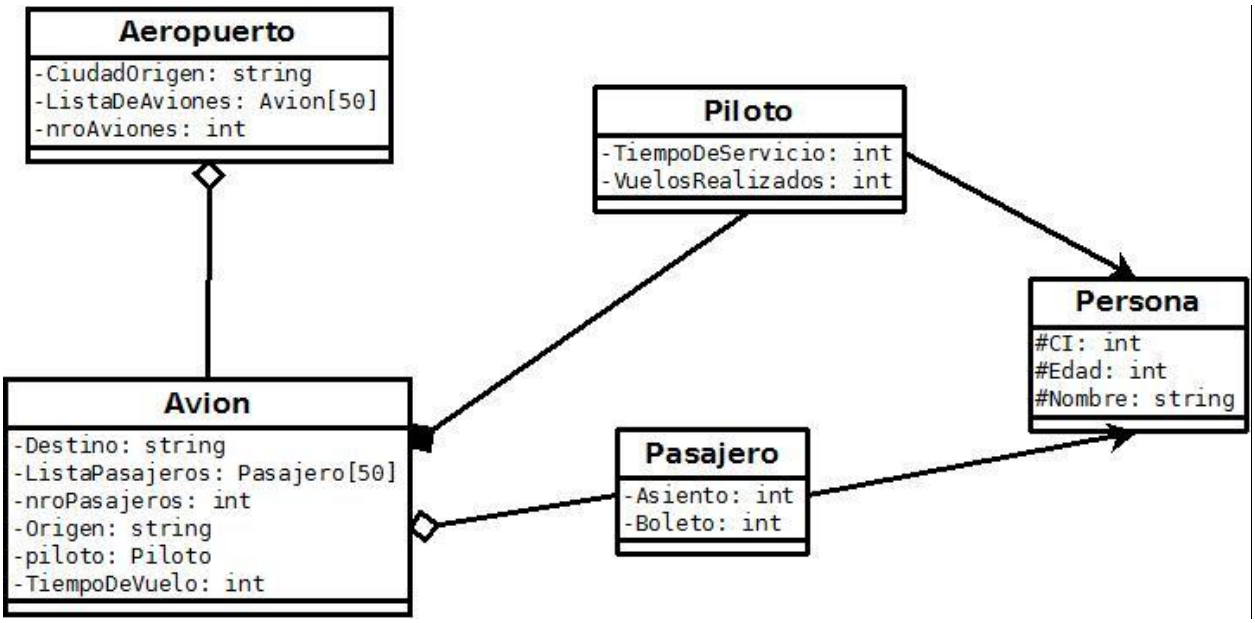
3. Sea el siguiente diagrama de clases:



a. Asignar como mínimo 2 atributos significativos a cada clase Hija.
b. Instanciar 1 objeto de cada clase de distinta manera.
c. Realizar un método que compare 3 muebles y muestre al más barato.
d. Mostrar el precio más barato en el que se puede conseguir un mueble, mostrar a su fabricante.
e. Para 3 objetos del tipo mueble, mostrar el objeto que se puede conseguir al precio más bajo

Laboratorio N° 4

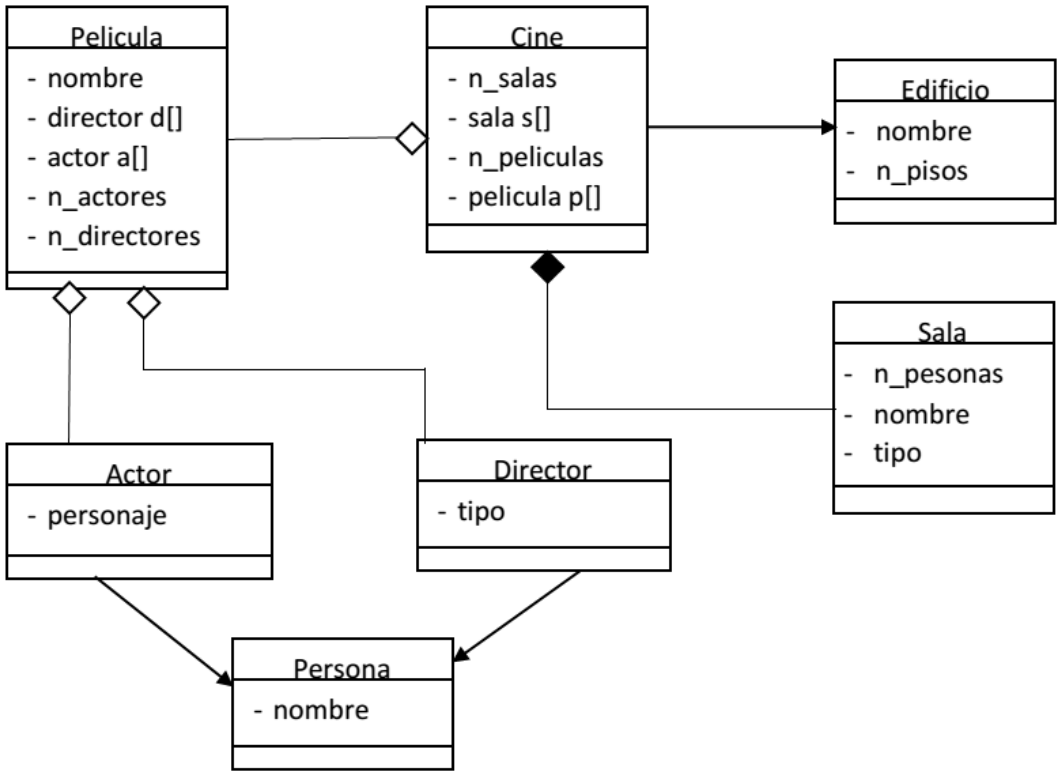
1. Sea el siguiente diagrama de clases:



   a) Instanciar 1 objeto de la clase Aeropuerto.
   b) Crear un método para adicionar pasajeros, cuidando que no se ocupe un asiento 2 veces.
   c) Ordenar a los aviones del aeropuerto según los vuelos realizados por cada piloto.
   d) Comparar 2 aviones, y si tienen el mismo origen y destino, juntar a sus pasajeros en un avión.

2. Una materia tiene como máximo 4 Paralelos, un paralelo consta de un docente y un vector de estudiantes. Un estudiante solo puede estar inscrito a un solo paralelo de una materia. Tanto docentes como estudiantes son personas, un docente puede estar asignado a uno o varios paralelos. Un paralelo además tiene un horario determinado y un aula.
   a. Crear el diagrama de clases para dicho enunciado, además de crear métodos para lo siguiente:
   b. Adicionar un estudiante al paralelo X de la materia Y, cuidando de que el estudiante no esté inscrito en algún otro paralelo de la misma materia.
   c. Cambiar al paralelo X, de la materia Y al estudiante con matricula Z.
   d. Ordenar a todos los estudiantes por edad, considerar dividirlos por paralelos.

3. Dado la siguientes relaciones entre las clases resolver los siguientes incisos:



a. Mostrar la película con el nombre z y el actor x
b. Mostrar la película con el director x y del tipo z
c. Mostrar la película con mayor cantidad de actores
d. Mostrar la sala con el mayor número de personas
e. Mostrar todas las salas del tipo x

Laboratorio N° 5

1. Dado el archivo *celulares.dat* que tiene la siguiente estructura:

Código	Propietario	Numero Celular	Código IMEI
1	Gustavo Jiménez	77509575	358987010052195
2	Katherine Vásquez	78978845	558784656905455
3	Juan Pérez Solís	60155451	735667435235566
4	Gabriela Guzmán	79879855	436545454545424
…	…	…	…

Dado el archivo *listaNegra.dat* que tiene la siguiente estructura:

Código IMEI	Motivo
358987010052195	Robo del teléfono
735667435235566	Extravió del teléfono
…	…

Se pide

a) Eliminar los registros de celulares.dat que estén en la listanegra.dat
b) Dividir el archivo celulares.dat en tres distintos archivos de la siguiente forma (Considerar los primeros 3 dígitos para identificar su proveedor):
c) En *entel.dat*, todos los números de Entel
d) En *viva.dat*, todos los números de Viva
e) En *tigo.dat*, todos los números de Tigo
f) Mover los primeros k registros el archivo *entel.dat al archivos viva.dat*.

2. Sea el archivo cliente.dat, y venta.dat con las siguientes estructuras:

Cliente.dat

Código	Nombre	Teléfono	Dirección
100	Luis	71584521	Av. Las Palmas
101	Miguel	66587412	Av. Buenos Aires
102	Pedro	72549781	Av. Brasil

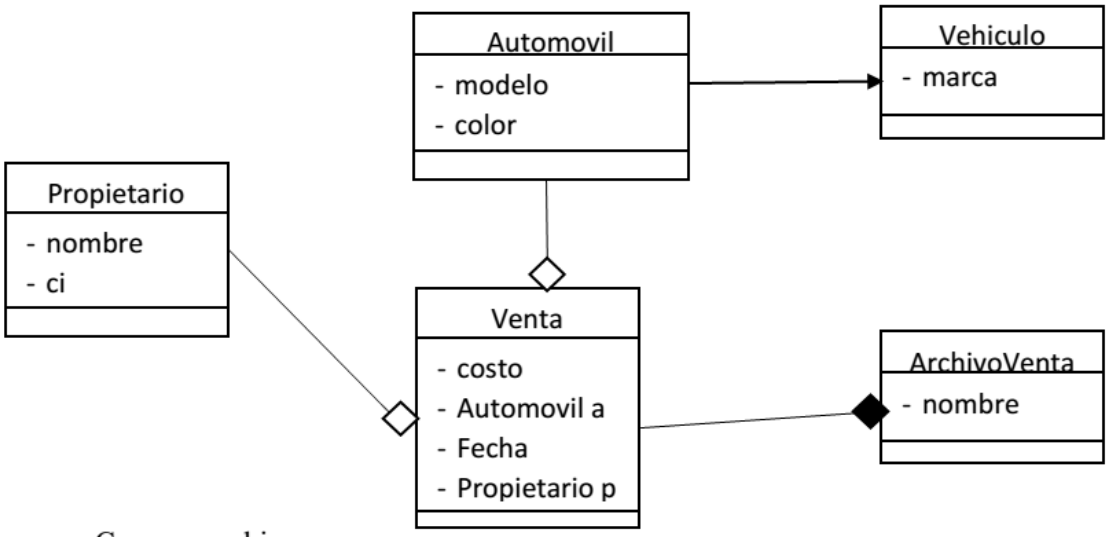
Venta.dat

Código	Producto	Cantidad	Monto	Fecha	Cliente
200	Calaminas	20	2000	12/02/2013	101
201	Ladrillos	200	1000	03/03/2013	100
202	Cemento	2	200	05/03/2013	102

a) Realizar el diagrama de clases

b) Mostrar la fecha de mayor ingreso en ventas
c) Cambiar el teléfono del cliente x por 72015741
d) Eliminar el cliente x y todas las ventas que haya realizado

3. Dado una empresa importadora de automóviles se obtiene el siguiente diagrama de clase



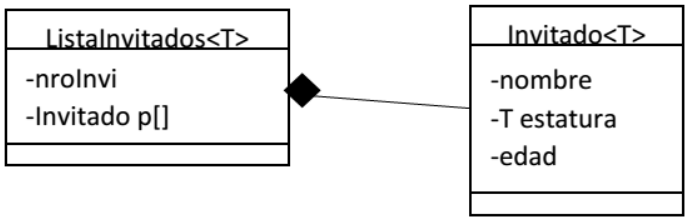
a. Crear un archivo
b. Hacer un menú
c. Obtener la lista de automóviles vendidos en la fecha x
d. Listar los vehículos vendidos al propietario x
e. Obtener el costo total de la empresa de automóviles

Laboratorio N° 6

1. Implementar La clase vector genérico que almacene 4 tipos de datos distintos <T,U,V,W>, se pide:
    a. ordenar ascendentemente el vector según el primer dato T,
    b. ordenar descendentemente según el tipo de dato U
    c. eliminar los datos repetidos según los datos V
    d. según los datos de W encontrar el mayor

2. Dada La clase matriz de conjuntos, donde los conjuntos pueden contener datos enteros, cadenas, caracteres, etc.
    a. Se pide modelar el diagrama de clases correspondiente
    b. Mostrar el conjunto que contenga más datos.
    c. Determinar si existe el elemento X en algún conjunto.

3. Dado el siguiente diagrama



    a. Mostrar a los menores de edad
    b. Mostrar a los de estatura mediana
    c. Mostrar a los de estatura 1,7