Implemente el constructor de la clase Fecha, de acuerdo a la documentación en la clase

* public Fecha (int anio, int mes, int dia, int hora, int minuto)

\*\*

\* Complete la clase segun lo indicado en el constructor

\*/

public class Fecha

{

private int anio;

private int mes;

private int dia;

private int hora;

private int minuto;

/\*\*

\* Lanza IllegalArgumentException cuando la fecha es invalida.

\* Anio debe ser mayor a 2010.

\* Hora en formato 24hs.

\* Validar mes y dia (no considera anios bisiestos. Ej, 29 FEB es invalido)

\*

\* Ej. fechas invalidas: 1999,12,12,12,12 ; 2010,2,29,21,6; 2012,1,31,24,01

\*

\* @param anio

\* @param mes

\* @param dia

\* @param hora

\* @param minuto

\*/

public Fecha (int anio, int mes, int dia, int hora, int minuto){

//TODO Implementar!!!!

}

public int getAnio() {

return anio;

}

public int getMes() {

return mes;

}

public int getDia() {

return dia;

}

public int getHora() {

return hora;

}

public int getMinuto() {

return minuto;

}

public String toString(){

StringBuilder sb = new StringBuilder();

sb.append(anio);

sb.append("/");

if (mes<10)

sb.append("0");

sb.append(mes);

sb.append("/");

if (dia<10)

sb.append("0");

sb.append(dia);

sb.append(" ");

if (hora<10)

sb.append("0");

sb.append(hora);

sb.append(":");

if (minuto<10)

sb.append("0");

sb.append(minuto);

return sb.toString();

}

}

Complete los métodos de la clase Llamada, de acuerdo a la documentación en la clase

* public String getDuracion();
* public String toString();

public class Llamada {

private TipoDeLlamada saliente;

private int duracionEnSegundos;

private String numero ;

private Fecha fecha;

/\*\*

\* Constructor con parametros.

\* @param numero

\* @param fecha

\* @param duracion

\* @param saliente

\*/

public Llamada (String numero, Fecha fecha, int duracion, TipoDeLlamada saliente){

if (numero == null || numero.trim().length() != 10)

throw new IllegalArgumentException();

if (duracion < 0)

throw new IllegalArgumentException();

this.numero = numero.trim();

this.fecha = fecha;

this.duracionEnSegundos = duracion;

this.saliente = saliente;

}

public TipoDeLlamada getTipoDeLlamada() {

return saliente;

}

public int getDuracionEnSegundos() {

return duracionEnSegundos;

}

public String getNumero() {

return numero;

}

public Fecha getFecha() {

return fecha;

}

/\*\*

\* Retorna la duracion en un string con formato <mm:ss>

\* Ejemplos: "03:15", "12:03", "110:45"

\*

\* @return

\*/

public String getDuracion() {

//TODO: Implementar!!!

return null;

}

/\*\*

\* Retorna un String con el siguiente fomrato:

\*

\* <aaaa/mm/dd> <hh:mm> <(SALIENTE)|(ENTRANTE)> <numero> <duracion>

\*

\* Ej. "2013/05/06 21:44 3514801234 ENTRANTE 05:12"

\* "2013/01/30 01:14 1161293091 SALIENTE 12:00"

\*

\*/

public String toString(){

//TODO: Implementar!!!

return null;

}

}

Implemente la clase RegistroDeLlamadas como una subclase de Registro, de acuerdo con la documentacion de la superclase.

* abstract public List<Llamada> getLlamadasPorDuracion (int minInicial, int segInicial, int minFinal, int segFinal) throws ParametrosInconsistentesException;
* abstract public List<Llamada> getLlamadasPorTipo (TipoDeLlamada tipo);
* abstract public int getDuracionTotalLlamadas ();
* abstract public Map<String, EstadisticasNumero> getMapaDeEstadisticas (List<Llamada> llamadas) throws ParametrosInconsistentesException;
* abstract public EstadisticasNumero getEstadisticasPorNumero (List<Llamada> llamadas, String numero) throws ParametrosInconsistentesException;

/\*\*

\* Complete la clase segun lo indicado en la superclase Abstracta

\*/

public class RegistroDeLlamadas extends Registro

{

// TODO Implementar

}

Implemente la clase Visor, que implementa la interfaz Formateable, de acuerdo a la documentación en la interfaz:

* public String convertirListaComoString(List llamadas);
* public String convertirListaComoStringAmigable(List llamadas, LibretaDeContactos cont);
* public String convertirMapaEstadisticaComoString(Map<String, EstadisticasNumero> m);

/\*\*

\* Complete la clase segun lo indicado en la interfaz

\*/

public class Visor implements Formateable {

// TODO Implementar

}