**PROBLEMA 5:**

La ***Universidad Técnica Particular de Loja*** organiza anualmente un congreso científico multidisciplinario internacional. Para la próxima edición, se requiere automatizar la gestión de las ponencias y participantes. Cada ponencia debe ser registrada con sus respectivos datos, y podrá tener múltiples participantes, quienes pueden ser ponentes o asistentes.

Es por lo que, el Doctor Santiago Acosta, Rector de la **UTPL**, le solicita desarrollar una aplicación que permita registrar las ponencias que se presentarán en el evento, incluyendo información como título, resumen, fecha programada, duración estimada, entre otros. Además, debe permitir vincular a los participantes implicados en cada ponencia. De cada participante se debe registrar su nombre completo, tipo de participación *(ponente o asistente)*, institución de procedencia, etc.

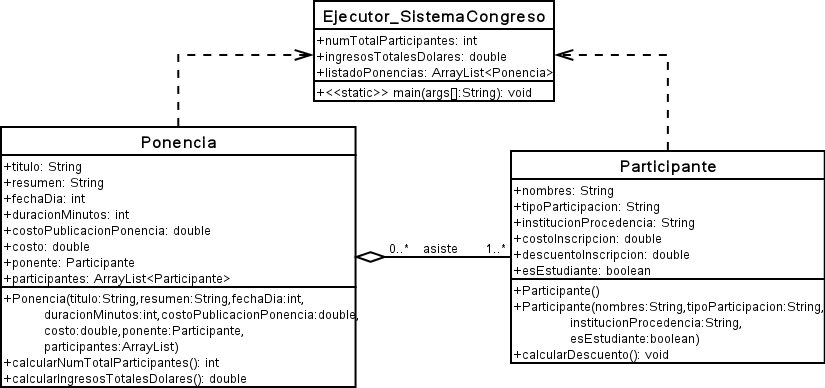
El sistema también debe incluir funcionalidades para:

* Automatizar el proceso de cobros por concepto de inscripción:
  + Como inscripción, los ponentes cancelan únicamente los costos por concepto de la publicación de su ponencia.
  + Los asistentes que pertenecen a la ***UTPL*** cancelar el **50%** del costo normal, y si es estudiante, únicamente el **10%**.
* Calcular el número total de participantes y los ingresos en dólares por ponencia.
* Generar un listado de ponencias, pero únicamente de ponentes extranjeros *(institución distinta a UTPL)*.

Reglas y consideraciones:

* El modelado debe establecer claramente las relaciones entre clases, aplicando agregación, cardinalidad, etc. en **UML**.
* La solución debe implementar estructuras de datos como arreglos estáticos y/o dinámicos.
* La entrada de datos puede simularse *(quemarse en código como fijas*) o registrarse desde consola.
* Se debe aplicar encapsulación, abstracción *(como pilares de la* ***POO****)* correctamente en todas las clases.
* El diseño orientado a objetos debe presentarse en formato **UML** y posterior su implementación en el lenguaje de alto nivel orientado a objetos **Java**.

**SOLUCIONARIO PROBLEMA 5:**

****

**import** **java.util.ArrayList**;

**import** **java.util.Arrays**;

**public** **class** **Ejecutor\_SistemaCongreso** {

**public** **static** **int** numTotalParticipantes; //Número total de participantes

**public** **static** **double** ingresosTotalesDolares; //Ingresos totales en dolares por ponencia

**public** **static** ArrayList<Ponencia> listadoPonenciasExternas; //Listado de ponencias externas a UTPL

**public** **static** **void** **main**(String[] args) {

Participante participanteDocente1 = **new** Participante("Juan Perez", "Participante", "UID", **false**);

Participante participanteDocente2 = **new** Participante("Maria Lopez", "Participante", "UMAN", **false**);

Participante participanteEstudiante1 = **new** Participante("Jorge Ruiz", "Participante", "UTPL", **true**);

Participante participanteEstudiante2 = **new** Participante("Luis Suquilanda", "Participante", "UNL", **true**);

Participante participantePonente1 = **new** Participante("Patricio Sisalima", "Ponente", "UMAN", **false**);

Participante participantePonente2 = **new** Participante("Mauricio Carrion", "Ponente", "UTPL", **false**);

ArrayList<Participante> listaParticipantesPonencia1 = **new** ArrayList(Arrays.asList(participanteDocente1, participanteDocente2, participanteEstudiante1, participanteEstudiante2));

ArrayList<Participante> listaParticipantesPonencia2 = **new** ArrayList(Arrays.asList(participanteDocente1, participanteDocente2));

Ponencia ponencia1 = **new** Ponencia("Peligros IA", "Este es mi resumen", **5**, **45**, **60**, **60**, participantePonente1, listaParticipantesPonencia1);

Ponencia ponencia2 = **new** Ponencia("Guerras y religión", "Este es mi resument", **6**, **40**, **50**, **50**, participantePonente2, listaParticipantesPonencia2);

listadoPonenciasExternas = **new** ArrayList<Ponencia>();

**if** (!ponencia1.ponente.institucionProcedencia.equals("UTPL"))

listadoPonenciasExternas.add(ponencia1);

**if** (!ponencia2.ponente.institucionProcedencia.equals("UTPL"))

listadoPonenciasExternas.add(ponencia2);

System.out.println("Total participantes Ponencia 1: " + (numTotalParticipantes = ponencia1.calcularNumTotalParticipantes() ));

System.out.println("Total participantes Ponencia 2: " + (numTotalParticipantes = ponencia2.calcularNumTotalParticipantes()) );

System.out.println("Total ingresos Ponencia 1: " + (ingresosTotalesDolares = ponencia1.calcularIngresosTotalesDolares() + ponencia1.ponente.costoInscripcion));

System.out.println("Total ingresos Ponencia 2: " + (ingresosTotalesDolares = ponencia2.calcularIngresosTotalesDolares() + ponencia2.ponente.costoInscripcion));

System.out.println("Ponencias externas");

**for** (Ponencia ponencia : listadoPonenciasExternas){

System.out.println(ponencia);

}

}

}

**class** **Ponencia**{

**public** String titulo; //Titulo de la ponencia

**public** String resumen; //Resumen de la ponencia

**public** **int** fechaDia; //Dia entero de la ponencia 1-31

**public** **int** duracionMinutos; //Duración de ponencia en minutos 0-N

**public** **double** costoPublicacionPonencia; //Costo en dolares por gastos de publicación

**public** **double** costo; //Costo de la inscripción en dolares para participantes

**public** Participante ponente;

**public** ArrayList<Participante> participantes; //Listado de participantes por Ponencia

**public** **Ponencia**(String titulo, String resumen, **int** fechaDia, **int** duracionMinutos, **double** costoPublicacionPonencia, **double** costo, Participante ponente, ArrayList<Participante> participantes) {

**this**.titulo = titulo;

**this**.resumen = resumen;

**this**.fechaDia = fechaDia;

**this**.duracionMinutos = duracionMinutos;

**this**.costoPublicacionPonencia = costoPublicacionPonencia;

**this**.costo = costo;

**this**.ponente = ponente;

ponente.costoInscripcion = **this**.costoPublicacionPonencia;

**this**.participantes = participantes;

}

**public** **int** **calcularNumTotalParticipantes**(){

**return** participantes.size();

}

**public** **double** **calcularIngresosTotalesDolares**(){

**double** ingresosTotalesDolares = **0**;

**for** (Participante participante : participantes) {

participante.costoInscripcion = **this**.costo;

participante.calcularDescuentos();

participante.costoInscripcion -= (participante.costoInscripcion \* (participante.descuentoInscripcion / **100**));

ingresosTotalesDolares += participante.costoInscripcion;

}

**return** ingresosTotalesDolares;

}

**@Override**

**public** String **toString**() {

**return** "Ponencia{" + "titulo=" + titulo + ", resumen=" + resumen + ", fechaDia=" + fechaDia + ", duracionMinutos=" + duracionMinutos + ", costoPublicacionPonencia=" + costoPublicacionPonencia + ", costo=" + costo + ", ponente=" + ponente + ", participantes=" + participantes + '}';

}

}

**class** **Participante**{

**public** String nombres; //Nombre completo de participantes

**public** String tipoParticipacion; //Ponente, Asistente

**public** String institucionProcedencia; //Institución de procedencia: Interna=UTPL, o externa

**public** **double** costoInscripcion; //Costo normal de la inscripcion

**public** **double** descuentoInscripcion; //Si es UTPL cancela 50% = (100%-50%), si es estudiante 10% => (100%-10%) => 90% descuento

**public** **boolean** esEstudiante; //Si es estudiante TRUE

**public** **Participante**() { }

**public** **Participante**(String nombres, String tipoParticipacion, String institucionProcedencia, **boolean** esEstudiante) {

**this**.nombres = nombres;

**this**.tipoParticipacion = tipoParticipacion;

**this**.institucionProcedencia = institucionProcedencia;

**this**.esEstudiante = esEstudiante;

}

**public** **void** **calcularDescuentos**(){

**this**.descuentoInscripcion = (**this**.institucionProcedencia.equals("UTPL")) ? **50** : **0** ;

**this**.descuentoInscripcion = (**this**.esEstudiante) ? **90** : **0** ;

}

}