**CONTROL 2**

**Programación Orientada a Objeto, 2024-2**

Profesores: Ma. Antonieta Soto Ch. (S1), Miguel Romero V. (S2) Fecha: 25-09-2024

Estudiante: Tiempo: 11:10 a 13:00

La Figura 1 muestra la representación UML de un diagrama de clases parcial que permitiría implementar un sistema de matrícula a programas de formación de una institución de educación superior.

**Se pide**:

Implementar los elementos presentes en el diagrama de clases, tal y como allí aparecen.

Para lo anterior tome en consideración los antecedentes adicionales que se dan a continuación.

**a) Debe ajustarse al diagrama de clases.** No agregue atributos, métodos, constructores u otro elemento en su implementación. Solo puede agregar métodos privados, si ellos permiten simplificar, reutilizar o hacer más legible el código.

**b) Clase Diplomado**

Respecto de esta clase contemple lo siguiente:

* El costo de un diplomado es el 90% de la suma del costo de los cursos que lo conforman.
* El método getTipo retorna el nombre de la clase del objeto this con la primera letra en mayúscula y las restantes en minúscula.
* El método addCurso retorna verdadero si es posible agregar el curso que se indica al diplomado y falso en caso contrario.

**c) Clase Curso**

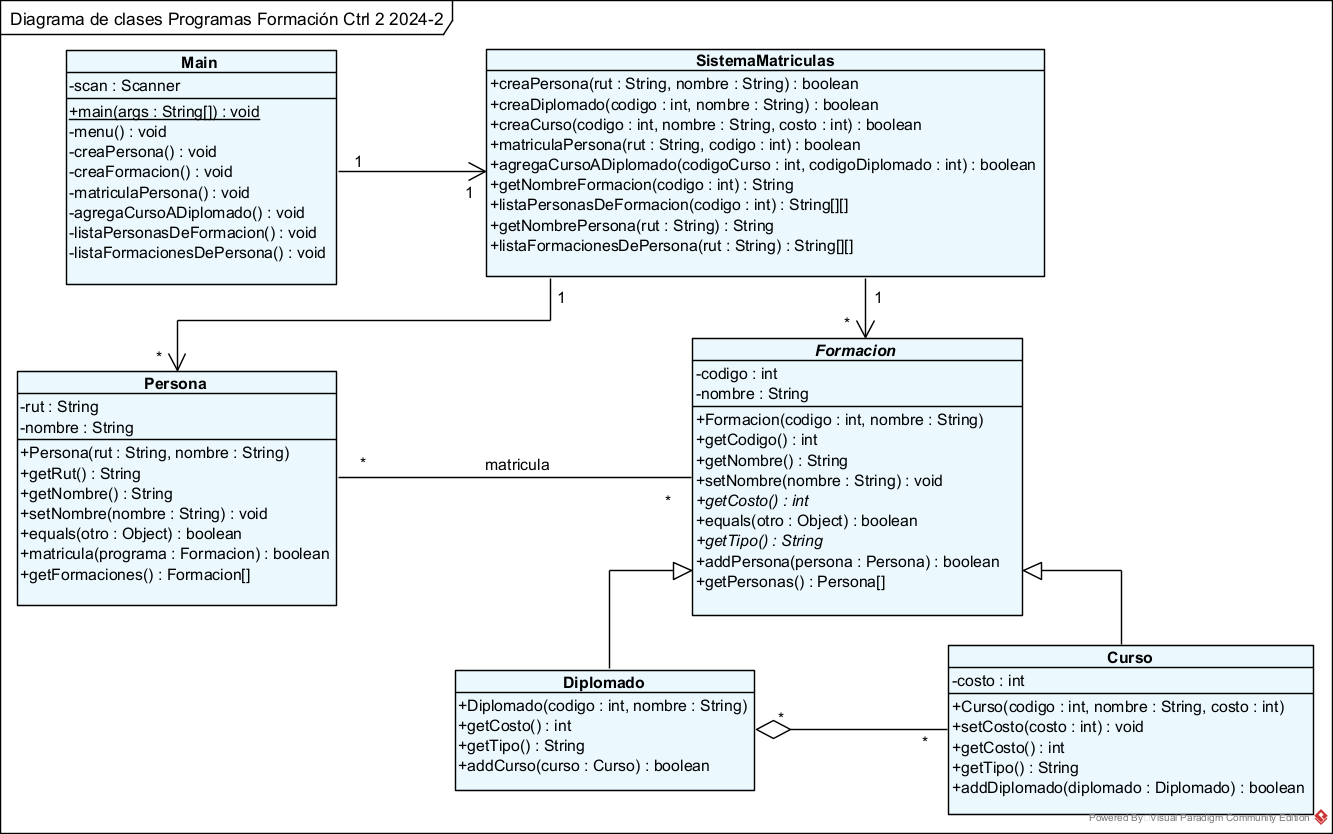
Respecto de esta clase contemple lo siguiente:

* El costo de un curso es el que registra el objeto this.
* El método getTipo retorna el nombre de la clase del objeto this con la primera letra en mayúscula y las restantes en minúscula.
* El método addDiplomado retorna verdadero si es posible asociar el diplomado que se indica al curso y falso en caso contrario.

**d) Clase Formación**

Respecto de esta clase contemple lo siguiente:

* Dos objetos Formacion son iguales si su código es igual. Lo anterior implica que no habrá dos objetos de tipo Formacion con igual código.
* El método addPersona retorna verdadero si es posible agregar la persona que se indica al programa de formación y falso en caso contrario.
* El método getPersonas retorna un arreglo con todas las personas que se han matriculado en el programa de formación this. Si no existen personas matriculadas aún, retorna un arreglo de tamaño cero.



*Figura 1. Diagrama de clases de un sistema de matrículas.*

**e) Clase Persona**

Respecto de esta clase contemple lo siguiente:

* Dos objetos Persona son iguales si su rut es igual.
* El método matricula retorna verdadero si es posible matricular el programa de formación que se indica a la persona y falso en caso contrario.
* El método getFormaciones retorna un arreglo con todos los programas de formación en los que se ha matriculado la persona this. Si la persona no se ha matriculado en ningún programa de formación aún, retorna un arreglo de tamaño cero.

**f) Clase SistemaMatriculas**

Respecto de esta clase contemple lo siguiente:

* El método getNombrePersona retorna el nombre de la persona cuyo rut se recibe como parámetro o null cuando no exista una persona con el rut que se indica.
* El método getNombreFormacion retorna el nombre del programa de formación cuyo código se recibe como parámetro o null cuando no exista un programa de formación con el código que se indica.
* El método creaPersona retorna verdadero si es posible crear una persona con los datos que se reciben como parámetro porque no existe otra persona con el rut indicado y falso en caso contrario.
* El método creaDiplomado retorna verdadero si es posible crear un diplomado con los datos que se reciben como parámetro porque no existe otro programa de formación con el código indicado y falso en caso contrario.
* El método creaCurso retorna verdadero si es posible crear un curso con los datos que se reciben como parámetro porque no existe otro programa de formación con el código indicado y falso en caso contrario.
* El método matriculaPersona retorna verdadero si es posible matricular la persona cuyo rut se recibe como parámetro en el programa de formación cuyo código se recibe como parámetro y falso en caso contrario. No será posible matricular una persona en un programa de formación si no existe una persona con el rut indicado o no existe un programa de formación con el código indicado o porque la persona ya se ha matriculado en el programa de formación.
* El método agregaCursoADiplomado retorna verdadero si es posible agregar el curso cuyo código se recibe como parámetro al diplomado cuyo código se recibe como parámetro y falso en caso contrario. No será posible agregar un curso a un diplomado si no existe un curso con el código indicado o no existe un diplomado con el código indicado o porque el curso ya ha sido agregado al diplomado.
* listaPersonasDeFormacion retorna un arreglo bidimensional con los datos de todas las personas que se han matriculado en el programa de formación cuyo código se recibe como parámetro. Cada fila almacena los datos de una persona. Los datos son rut y nombre. Si no existen personas matriculadas en el programa de formación o no existe un programa de formación con el código dado, el método retorna un arreglo de tamaño cero.
* listaFormacionesDePersona retorna un arreglo bidimensional con los datos de todos los programas de formación en los que se ha matriculado la persona cuyo rut se recibe como parámetro. Cada fila almacena los datos de un programa de formación. Los datos son código, nombre, tipo y costo. Si la persona no se ha matriculado aún en algún programa de formación o no existe una persona con el rut dado, el método retorna un arreglo de tamaño cero.

**g) Clase Main**

Respecto de esta clase considere que:

* El método main (único elemento estático de la clase) debe invocar el método menu.
* El método menu debe desplegar el menú de opciones, leer la opción escogida por el usuario e invocar el método privado de la propia clase que se encargará realizar lo solicitado por el usuario.
* El método creaPersona debe leer el rut y nombre de la persona a crear e invocar el método apropiado de la clase SistemaMatriculas que permita crear la nueva persona. El método debe desplegar un mensaje informando al usuario que la operación se realizó con éxito o que no ha sido posible realizar lo solicitado, según corresponda.
* El método creaFormacion debe leer, en primer lugar, el tipo de programa de formación que se desea crear (diplomado o curso). Tratándose de un diplomado se deberá leer, además, el código y nombre. Tratándose de un curso se deberá leer código, nombre y costo. A continuación, se debe invocar el método apropiado de la clase SistemaMatriculas que permita crear el curso o diplomado. El método debe desplegar un mensaje informando al usuario que la operación se realizó con éxito o que no ha sido posible realizar lo solicitado, según corresponda.
* El método matriculaPersona debe leer el rut de la persona a matricular y el código del programa de formación en el que se desea matricular. Luego, debe invocar el método apropiado de la clase SistemaMatriculas que permita matricular la persona en el programa de formación correspondiente. El método debe desplegar un mensaje informando al usuario que la operación se realizó con éxito o que no ha sido posible realizar lo solicitado, según corresponda.
* El método agregaCursoADiplomado debe leer el código del curso y el código del diplomado en el que se desea agregar el curso. Luego, debe invocar el método apropiado de la clase SistemaMatriculas que permita agregar el curso al diplomado. El método debe desplegar un mensaje informando al usuario que la operación se realizó con éxito o que no ha sido posible realizar lo solicitado, según corresponda.
* El método listaPersonasDeFormacion debe leer el código del programa de formación cuyos estudiantes matriculados se desean desplegar. Luego, debe invocar los métodos apropiados de la clase SistemaMatriculas que permitan obtener los datos necesarios y desplegar un listado como el que se muestra en la Figura 2. En el título aparece el nombre del programa de formación.

Si no existen datos, se debe desplegar un mensaje informando al usuario.

Listado Personas de Programa de Formación “XXXXXXXXXXXXXXX”

Rut Nombre

99.999.999-X XXXXXXXXXXX

….

Nro. total estudiantes matriculados: 999

*Figura 2. Listado de personas matriculadas en un programa de formación.*

* El método listaFormacionesDePersona debe leer el rut de la persona cuyos programas de formación en los que se ha matriculado se desea desplegar. Luego, debe invocar los métodos apropiados de la clase SistemaMatriculas que permitan obtener los datos necesarios y desplegar un listado como el que se muestra en la Figura 3. Note que Tipo será Curso o Diplomado. En el título aparece el nombre de la persona.

Si no existen datos, se debe desplegar un mensaje informando al usuario.

Listado Programa de Formación de Persona “XXXXXX XXXXXXXXX”

Código Nombre Tipo Costo ($)

999999 XXXXXXXXXXX XXXXXXXXX 99.999.999

….

Monto total programas: $999.999.999

*Figura 3. Listado de programas de formación de una persona.*

**Importante**:

* Suponga que los datos leídos son siempre válidos, por lo que no necesita hacer validaciones, salvo las situaciones que se haya indicado expresamente en el enunciado que debe contemplar.
* Las copias desde cualquier fuente serán evaluadas con nota 1.0. Lo mismo aplica a quienes faciliten la copia.