PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 2

1era. práctica (tipo b)
(Segundo Semestre 2022)

Indicaciones Generales:

- Tiempo estimado: 1h 50 minutos
- Se les recuerda que, de acuerdo al reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de esta práctica.
- Está permitido el uso de apuntes de clase, diapositivas, ejercicios de clase y código fuente. (Debe descargarlos antes de iniciar con la solución del enunciado)
- No está permitido el uso de entornos de desarrollo integrado (IDEs).
  Debe utilizar un editor de texto: Notepad++, Sublime, etc.
- Está permitido el uso de Internet (únicamente para consultar páginas oficiales de Microsoft y Oracle). No obstante, está prohibida toda forma de comunicación con otros estudiantes o terceros.

PARTE PRÁCTICA (20 puntos)

PUEDE UTILIZAR MATERIAL DE CONSULTA.
Se considerará en la calificación el uso de buenas prácticas de programación (aquellas vistas en clase).

PREGUNTA 1 (20 puntos)

Actualmente nuestra universidad cuenta con un conjunto de bibliotecas que se encuentran disponibles para la comunidad PUCP. Dentro de estas bibliotecas, existen espacios de trabajo conocidos como “cubículos de estudio” que pueden ser utilizados tanto por profesores, como estudiantes y egresados para establecer reuniones de trabajo o discusión, compartir ideas o inclusive para avanzar tareas, proyectos o cualquier actividad académica que sea grupal. Si bien se cuenta además con un sistema de reservas de estos “cubículos de estudio”, se desea reestructurar todo este sistema por lo que se le ha contactado a Ud. para la programación de toda la lógica del negocio empleando cualquiera de los lenguajes: JAVA o C#.

Con respecto a las bibliotecas de la universidad, se desea registrar el nombre y un indicador que permita saber si se trata o no de una biblioteca ubicada dentro del campus universitario, ya que la PUCP también dispone de bibliotecas ubicadas fuera del campus, como es el caso de la “*Biblioteca del Instituto Confucio*” y la “*Biblioteca del Instituto Riva-Agüero*”. Con respecto a un cubículo, este posee un nombre, el número de piso en el que está ubicado dentro del edificio de la biblioteca, una capacidad máxima de personas que pueden ingresar al mismo, un indicador para saber si el cubículo dispone de televisor, un indicador para saber si dispone de pizarra, un indicador para saber si dispone de proyector y un indicador para saber si dispone de computador.

Como se mencionó, estos cubículos pueden ser reservados tanto por profesores como estudiantes y egresados que vienen a ser miembros de la comunidad universitaria PUCP.
Todos estos miembros de la comunidad PUCP comparten características en común como un nombre, un apellido paterno y un código PUCP. Sin embargo, los profesores además poseen una categoría (que puede ser cualquiera de estos cuatro tipos: Contratado, Auxiliar, Asociado o Principal). Asimismo, los estudiantes deben tener un indicador que permita saber si están matriculados o no en el semestre actual. Por último, de un egresado se necesita saber además si es o no titulado.

Un cubículo puede reservarse varias veces, y es por ello que está relacionado a múltiples reservas. Para la reserva de un cubículo es necesario establecer la fecha en la cual será requerido, así como la hora de inicio y la hora fin de uso de este. (Por ejemplo, se podría reservar el cubículo *2-03* de la *Biblioteca Complejo de Innovación Académica* para el día *25 de agosto del 2022* de *10:00* a *11:00*). Una reserva es realizada por un grupo de miembros de la comunidad PUCP, entre los cuales podrían estar profesores y/o estudiantes y/o egresados. Cada vez que se genere una reserva, el programa deberá generar un número entero único que sea correlativo con el que será posible identificar a la misma.

Se requiere que se puedan consultar los datos de todos los miembros de la comunidad PUCP que participen de estas reservas. En este sentido, si se consultan los datos de un profesor, se devolverá la cadena “Profesor:”, seguido del código PUCP, el nombre, el apellido paterno y la categoría. Asimismo, si se consultan los datos de un estudiante, se devolverá la cadena “Estudiante:” seguido del código PUCP, el nombre, el apellido paterno y el indicador para saber si está o no matriculado actualmente. Finalmente, si es que se consultan los datos de un egresado, se devolverá la cadena “Egresado:” seguido del código PUCP, el nombre, el apellido paterno, el indicador para saber si tiene título universitario.

Una reserva también será consultable. En este caso, al consultar los datos de una reserva se espera obtener su id, seguido de la hora de inicio, la hora fin y los responsables de esa reserva.

Teniendo en cuenta todo lo mencionado, se ha realizado un análisis preliminar y se han detectado las siguientes clases, las cuales deben considerarse y por ningún motivo podrá eliminar:

- **Biblioteca**: Clase que define a las bibliotecas de la universidad, que asimismo, definirá un método llamado "**`String consultarReservasdelCubiculo(String nombre, Date fecha)`**" que devolverá la información del cubículo con el nombre especificado y de todas las reservas asociadas al mismo para la fecha indicada.
- **Cubiculo**: Clase que define a los cubículos de una biblioteca, que, asimismo, definirá los métodos
  - **`void registrarReserva(Reserva reserva)`** que permitirá asociar una reserva al cubículo.
  - **`String consultarReservas(Date fecha)`** que devolverá la información de todas las reservas asociadas al cubículo en la fecha indicada.
- **IConsultable:** Clase de tipo interface, que define la obligación de consultar datos **("String consultarDatos()")** tanto para miembros de la comunidad PUCP como para las reservas.

Para validar el modelado de clases y como parte de las pruebas del sistema, se cuenta con la siguiente clase:

```
import java.util.ArrayList;
import java.time.LocalTime;
import java.text.SimpleDateFormat;
public class Principal{
	public static void main(String[] args) throws Exception{
		//Objeto SDF
		SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
		//Se crea una biblioteca
		Biblioteca biblioteca1 = new Biblioteca("Biblioteca Central",true);
		//Se crean dos cubículos de la Biblioteca Central
		Cubiculo cub01 = new Cubiculo("101",1,3,true,true,true,true);
		Cubiculo cub02 = new Cubiculo("311",3,3,false,true,false,false);
		//Se asignan los cubículos a la biblioteca
		biblioteca1.setCubiculos(new ArrayList<>());
		biblioteca1.getCubiculos().add(cub01);
		biblioteca1.getCubiculos().add(cub02);
		//Vamos a suponer que un profesor en conjunto con un estudiante y un egresado desean
reservar el cubículo 101
		//Se crea a los interesados
		MiembroPUCP persona1 = new Profesor("20022214","JUAN","ARENAS",Categoria.Auxiliar);
		MiembroPUCP persona2 = new Estudiante("20167441","ROSANGELA","VALENZUELA",true);
		MiembroPUCP persona3 = new Egresado("20140445","DANIELA","ARGUMANIS",true);
		//Generamos la reserva para hoy de 10:00 a 11:00
		Reserva reserva1 = new Reserva(sdf.parse("25-08-
2022"),LocalTime.of(10,00),LocalTime.of(11,00));
		//Asociamos a los responsables de esta reserva
		reserva1.setResponsables(new ArrayList<>());
		reserva1.getResponsables().add(persona1);
		reserva1.getResponsables().add(persona2);
		reserva1.getResponsables().add(persona3);
		//Inicializamos las reservas del cubiculo 101
		cub01.setReservas(new ArrayList<>());
		//Asociamos la reserva al cubiculo 101
		cub01.registrarReserva(reserva1);
		//Vamos a suponer que un estudiante desea reservar el cubiculo 101 para hoy también pero de
11:00 a 12:00
		MiembroPUCP persona4 = new Estudiante("20135096","ESTEFANI","SILVA",true);
		Reserva reserva2 = new Reserva(sdf.parse("25-08-
2022"),LocalTime.of(11,00),LocalTime.of(12,00));
		reserva2.setResponsables(new ArrayList<>());
		reserva2.getResponsables().add(persona4);
		cub01.registrarReserva(reserva2);
		//Ahora consultamos las reservas del cubiculo 101 de la biblioteca para hoy
		String reporte = biblioteca1.consultarReservasDelCubiculo("101",sdf.parse("25-08-2022"));
		System.out.println(reporte);
	}
}
```

La ejecución del programa debe generar la siguiente salida:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PC_Lima\OneDrive\LP2\2022-2\Laboratorio01\Solucion>javac *.java

C:\Users\PC_Lima\OneDrive\LP2\2022-2\Laboratorio01\Solucion>java Principal
Reservas del cubiculo 101 de la Biblioteca Central para 25-08-2022
Reserva 1: de 10:00 a 11:00
        Profesor(a): 20022214 - JUAN ARENAS - Auxiliar
        Estudiante: 20167441 - ROSANGELA VALENZUELA - Matriculado(a): true
        Egresado(a): 20140445 - DANIELA ARGUMANIS - Titulado(a): true
Reserva 2: de 11:00 a 12:00
        Estudiante: 20135096 - ESTEFANI SILVA - Matriculado(a): true


C:\Users\PC_Lima\OneDrive\LP2\2022-2\Laboratorio01\Solucion>
```

En las reservas, para la asignación de horas en JAVA debe utilizar **LocalTime (java.time.LocalTime)**. En el caso de C# utilizará **TimeSpan. Asimismo, para comparar fechas en JAVA utilice equals, por ejemplo:** **`reserva.getFecha().equals(fecha)`**

Es indispensable que se encuentren programados todos los atributos de las clases y sus relaciones (como mínimo la programación de las relaciones que permiten la impresión del reporte). Con respecto a los constructos, getters y setters, puede programar solo aquellos que son requeridos para la salida del reporte.

Se le solicita programar en JAVA o C# todas las clases que dan soporte lo mencionado en el caso, a la lógica del negocio y a la salida del reporte. Debe subir a PAIDEIA en un archivo .zip, todo el código fuente (archivos .java o archivos .cs).

Aspectos a considerar para evitar descuento de puntos:

- Nombrar correctamente a las clases, atributos y métodos.
- Utilice los principios de POO: abstracción (en clases y métodos donde sea requerido), polimorfismo y encapsulamiento.
- Utilice correctamente ámbitos, clases/métodos abstractas(os), clases de tipo interface, enumerados donde sea requerido.
- Colocar a modo de comentario su nombre completo y código PUCP en la parte superior de todas las clases.
- Respetar el orden en la estructura de una clase.
- Emplear un archivo por clase.
- El programa debe compilar correctamente, se descontarán puntos por errores de compilación.

Rúbrica de calificación:

(0.5 punto) Correcta programación de la clase de tipo interface.
(0.5 punto) Correcta programación de la clase enumerate.
(5 puntos) Correcta programación del método consultarDatos() en las clases donde es requerido el uso de la interface.
(1 punto) Correcta programación del método registrarReservar en Cubiculo.
(3 puntos) Correcta programación del método consultarReservas en Cubiculo.
(2 puntos) Correcta programación del método `consultarReservasDelCubiculo`.
(6 puntos) Correcta programación de todas las clases con sus atributos, relaciones, constructos, getters y setters (respetando encapsulamiento, abstracción (métodos y clases), herencia, interfaces y enumerados).
(2 puntos) Correcta programación de la clase Principal y visualización del reporte.

Profesor del Curso:
Dr. Freddy Paz

25 de agosto del 2022