Universidad Internacional de las Américas
Escuela de Ingeniería Informática
Informe del laboratorio realizado

Nombre del curso:	**Programación I**
Nombre del docente:	**Lic. Carlos González Romero**
Fecha actual:	**19/06/2023**
Fecha de entrega:	**19/06/2023 6:00PM**
Nombre del estudiante:	
Número de laboratorio:	**2**
Calificación:	**5%**

Nota importante:

- Estimado estudiante, si el informe de laboratorio usted lo envía fuera del tiempo solicitado, queda a criterio del docente aceptarlo y si se acepta será calificado sobre el 50% del valor del informe.
- La respuesta del laboratorio debe ser con lo visto en clase, de presentarse una solución cuya codificación no sea con lo visto en clase, el laboratorio queda automáticamente anulado.
- El estudiante debe cargar una carpeta [Apellido1Apellido2Nombre] con la entrega del proyecto programado solicitado en el laboratorio.
- La entrega del laboratorio es en la carpeta compartida con el estudiante, cuyo enlace tipo DriveGoogle se encuentra en la semana que corresponde en el E-Campus. El profesor al corte de entrega del laboratorio reubica los laboratorios entregados en otra carpeta donde el estudiante no tiene acceso y envía una imagen a todo el grupo para oficializar la entrega del mismo, un estudiante cuenta con 8 horas para indicar si no ve su laboratorio vía correo electrónico directo al profesor (gonzalezcarlos7684@gmail.com)

OBJETIVO GENERAL DEL LABORATORIO:

Parte #1 (1%)

- Indique dos ventajas de realizar la captura de errores e indique la estructura de código fuente Java requerida para aplicar esto.

  1.Manejo de excepciones: le permite identificar y controlar condiciones inusuales durante la ejecución del programa, evitando bloqueos y comportamientos inesperados cuando surgen condiciones inesperadas.
  2.Robustez y confiabilidad: al detectar y manejar correctamente los errores, los programas se vuelven más flexibles y confiables. Puede prever problemas potenciales y tomar las medidas adecuadas para manejarlos correctamente, aumentando su confiabilidad en diferentes escenarios.
  Ejemplo:

```
public class EjemploCapturaErrores {
   public static void main(String[] args) {
      // Bloque de código donde puede ocurrir el error
     try {
         int[] numeros = {1, 2, 3};
         System.out.println("El elemento en la posición 5 es: " + numeros[5]);
      // Código capturado  para manejar la excepción
      } catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Error: Acceso a un índice inválido del arreglo.");
         System.out.println("Mensaje de error: " + e.getMessage());
// Bloque opcional de código que se ejecuta siempre independientemente de si se produce una excepción o no
               } finally {
         System.out.println("Finalizando la ejecución del programa.");
      }
   }
}
```

- Explique 2 objetivos de Debuggiar nuestro código fuente Java

1. Identificar y corregir errores: El objetivo principal es encontrar y solucionar los errores presentes en el código para asegurarse de que el programa funcione como se espera y produzca los resultados deseados.

2.Optimizar el rendimiento: La depuración también se utiliza para identificar y mejorar las partes del código que pueden ser ineficientes, lo que permite optimizar el rendimiento del programa y garantizar su eficiencia en términos de velocidad y consumo de recursos.

Parte #2 (4%)

- Investigue sobre el concepto de Herencia en el contexto de programación orientada a objetos y mencione 3 beneficios de aplicar la aplicación de este concepto.

  La herencia en el contexto de la programación orientada a objetos es un mecanismo que permite crear nuevas clases basadas en clases existentes y establecer relaciones "es un" entre ellas. Las subclases heredan las propiedades y los métodos de la clase principal, lo que permite la reutilización del código y la organización de las clases en una jerarquía.

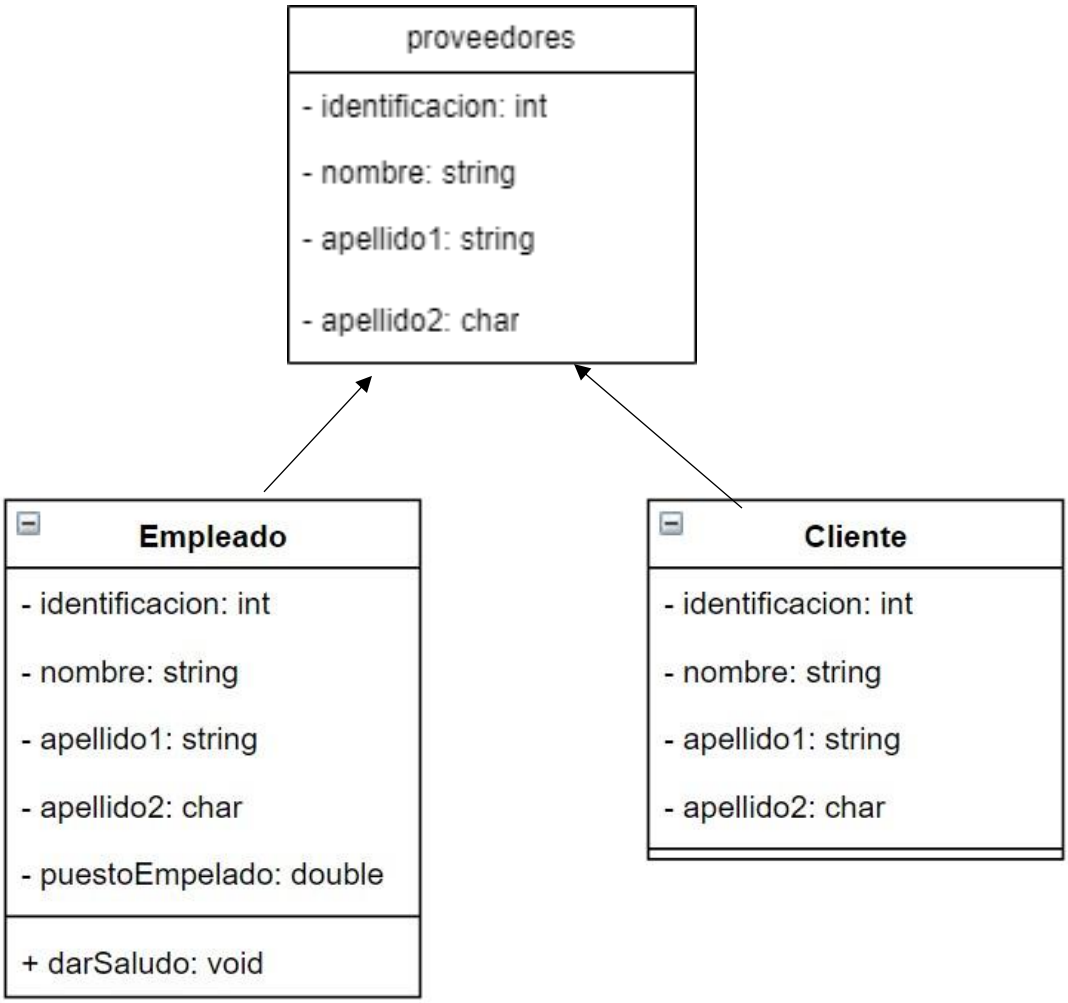
  1.Reutilización de código: La herencia permite aprovechar el código existente en una clase padre al crear una nueva clase hija, evitando la duplicación de código y promoviendo la eficiencia en el desarrollo.

  2.Extensibilidad y flexibilidad: La herencia permite agregar nuevos atributos y métodos en la clase hija para extender la funcionalidad de la clase padre, adaptándola a requisitos específicos. También permite crear múltiples clases hijas a partir de una clase padre, lo que brinda flexibilidad en la creación de nuevas clases y adaptación del diseño.

  3.Organización y jerarquía: La herencia organiza las clases en una jerarquía basada en la relación de "es un", lo que proporciona una estructura clara y coherente. Esto facilita la comprensión y el mantenimiento del código, además de permitir la categorización y agrupación lógica de las clases en el proyecto de software.

  - Escriba el codigo fuente del siguiente diseñe UML de clases donde se muestre la aplicación del concepto de atributos y metodos para las clases [Empleado] y [Cliente], ademas aplique el concepto de herencia entre las mismas.

- Se le ha contrato para desarrollar una pantalla de proveedores, diseñe el diagrama UML de clases que le permita poder incoporar al sistema el registro de proveedores sacando el maximo provecho del codigo fuente a traves del concepto de herencia, se requiere diagrama UML y codigo fuente que evidencia el concepto (utilizar como base el actual diagrama UML).



RECURSOS PARA USAR EN EL LABORATORIO:

- NetBeans
- Archivo txt para respuestas teóricas

ENTREGABLES

- Archivos .java versionar códigos de archivos JAVA en el respectiva carpeta con su nombre de Laboratorio#2 de GitHub