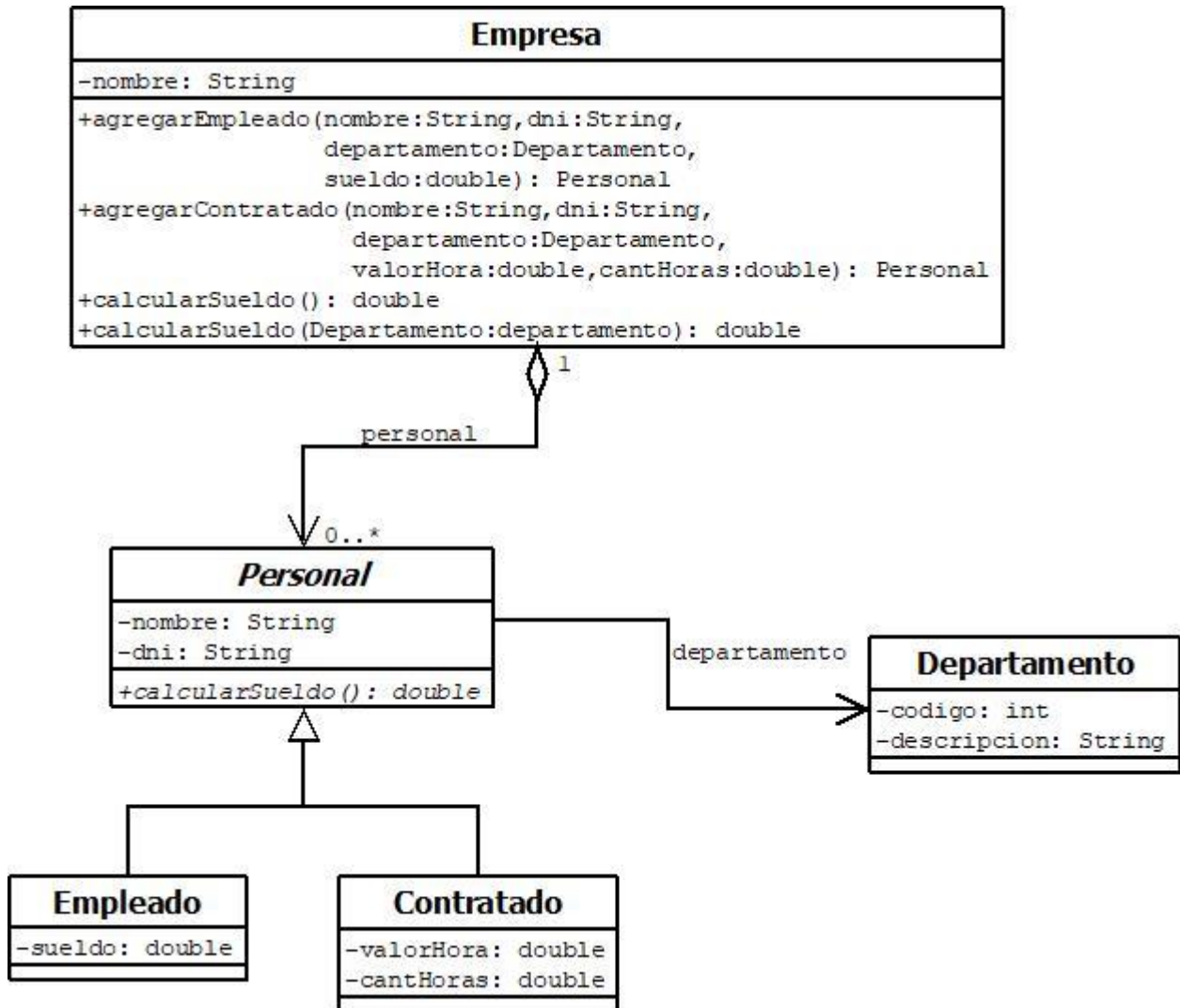


1º PARCIAL DE ALGORITMICA Y ESTRUCTURA DE DATOS II - 2020

1) Desarrollar un sistema que calcule los sueldos del personal de una Empresa. La misma tiene empleados permanentes y contratados. Los empleados permanentes cobran una suma fija (que puede ser distinta para cada uno) y los contratados cobran según su valor hora por la cantidad de horas trabajadas (que también puede ser distinta para cada uno). Todo trabajador dentro de la empresa pertenece a un departamento (por ejemplo: producción, ventas, compras, etc.)

Diagrama de Clases



Crear todas las clases con sus atributos y constructores. Implementar sus métodos

2) Realizar un programa de prueba que cree una empresa y cargar empleados fijos y contratados para distintos departamentos.

- Mostrar el importe a pagar en sueldos para todo el personal.
- Mostrar el importe a pagar en sueldos para un determinado departamento.

3) Dada una lista simplemente enlazada, agregar el siguiente método y realizar un programa que pruebe la misma y sus excepciones. Se evaluará su eficiencia.

```
/* Elimina (nro) elemento comenzando de la posición (pos) de la lista */
/* Si la acción no es válida lanza la excepción: IndexOutOfBoundsException */

/*
 * Por ejemplo dada la lista {A, B, C, D, E}
 * removePos(2,4) => {A, E}
 */
```

```

* removePos(1,2) => {C, D, E}
* removePos(5,1) => {A, B, C, D}
* removePos(5,2) => IndexOutOfBoundsException
* removePos(0,2) => IndexOutOfBoundsException
*
* El ejemplo muestra que elementos quedan en la lista original después de llamar
* al método removePos.
*/
public void removePos(int pos, int nro)

```

4) Dada una lista doblemente enlazada, agregar el siguiente método y realizar un programa que pruebe la misma y sus excepciones:

```

/* Retorna una nueva lista con los últimos (p) elementos de la lista */
/* Si la acción no es válida lanza la excepción: IndexOutOfBoundsException */

/*
* Por ejemplo dada la lista {A, B, C, D, E}
*
* rightList(2) => {D, E}
* rightList(1) => {E}
* rightList(5) => {A, B, C, D, E}
* rightList(0) => {}
* rightList(6) => IndexOutOfBoundsException
*
* Tener presente que este método retorna una lista nueva. No modifica la lista
* original.
*/
public DoublyLinkedList<E> rightList(int p) {

```

TEORIA

1. Analice las ventajas y desventajas que surgen de la comparación del uso de arreglos y listas simplemente enlazadas en las operaciones de:
 - Inserción ordenada
 - Eliminación ordenada
 - Búsqueda

Deberá tener en cuenta dos aspectos en cada caso: uso de memoria y velocidad de procesamiento del algoritmo.

2. ¿Cuál es la diferencia entre estudios experimentales de algoritmos y análisis de algoritmos? Ejemplifique.
3. ¿Qué provee el disponer de clases abstractas al tener que definir una jerarquía de clases?

Nota: se tendrá en cuenta la redacción correcta de las respuestas, es decir, que se entiendan los conceptos y que sintácticamente estén bien construidas.

IMPORTANTE:

1. Los enunciados no deberían dejar lugar a dudas de los ejercicios a resolver y preguntas a responder. De todas maneras si surge alguna consulta del enunciado enviar la misma al **Foro** que está en la **sección Evaluación**. Las preguntas serán respondidas dentro de los 30 minutos de realizadas.
2. Enviar un WhatsApp al grupo **solo** si tienen algún inconveniente (no pueden entrar al foro, no reciben la respuesta en el tiempo indicado, no pueden subir las soluciones, etc.)
3. Para la parte práctica del parcial, subir solamente los archivos con extensión .java.
4. Para la parte teórica del parcial, subir un solo archivo de Word. (con extensión doc o .docx)