

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

9na práctica (tipo b)
Segundo Semestre 2020

Indicaciones Generales:

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el audio de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear **variables globales**, **estructuras ni la clase String**. Tampoco se podrán emplear las funciones malloc, realloc, strdup o strtok, igualmente no se puede emplear cualquier función contenida en las bibliotecas stdio.h, cstdio o similares y que puedan estar también definidas en otras bibliotecas. **Tampoco podrá hacer uso de plantillas.**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada función **NO** debe sobrepasar las 20 líneas de código aproximadamente. El archivo **main.cpp** solo podrá contener la función **main** de cada proyecto y el **código contenido en él solo podrá estar conformado por tareas implementadas como funciones**. **En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos por archivo.**
- **NO SE CALIFICARÁN AQUELLAS FUNCIONES DESARROLLADAS EN EL MISMO ARCHIVO QUE LA FUNCIÓN main.**
- El código comentado **NO SE CALIFICARÁ.**
- Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Cada proyecto debe ser elaborado en un proyecto independiente, de no acatar esta restricción solo se calificará una pregunta.
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases así como guardar los estándares en la definición y uso de todas las clases desarrolladas.
- **Salvo en la sobrecarga de los operadores >> y <<, no se podrán definir funciones independientes que no estén ligadas como métodos a alguna de las clases planteadas.**
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.

SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .cpp O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

**NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES
DADAS EN LA PRUEBA**

Puntaje total: 20 puntos.

Cree una carpeta denominada "Lab09-2020-2", dentro de ella colocará los proyectos que a continuación se solicitan. **DEBE RESPETAR ESTE REQUERIMIENTO.**

Cuestionario:

En este laboratorio se desarrollará una aplicación que le permita manejar la atención dada a los pacientes que son asistidos en una institución de salud, por tal motivo se debe utilizar polimorfismo y listas ligadas para la solución a este requerimiento.

PREGUNTA 1 (10 puntos):

Se pide que desarrolle un proyecto denominado "Laboratorio09Preg01" en donde se declaren las clases descritas con las relaciones necesaria que permitan manipularlas empleando polimorfismo.

En la institución de salud pueden ser atendidos tres tipos de pacientes: ambulatorios, de urgencias y de emergencias. Estos tipos poseen datos comunes heredados de la clase Paciente, y también datos particulares en su propia clase. Por tal motivo se requiere desarrollar las siguientes clases como se muestra a continuación:

Paciente	Ambulatorio	Urgencia	Emergencia
int dni; char * nombre; char * codMed;	double porcSeguro;	double porcSeguro; double porcUrgencia;	int telefonoRef; char *nombreRef;

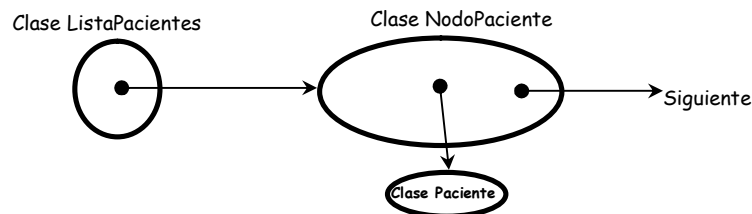
La implementación contemplará la definición de atributos, constructores por defecto, constructores copia, destructores, métodos selectores, un método que permita leer los datos desde un archivo de texto, otro que permita imprimirlo en otro archivo de texto, otro que devuelva el orden de impresión de acuerdo con el tipo de paciente (1: Ambulatorio, 2: Urgencia 3: Emergencia) y otro que indique si el paciente uso o no el seguro para su atención. La memoria asignada a las cadenas debe ser óptima.

La función main deberá probar, de manera muy sencilla, la implementación de estas clases y selectores desarrollados.

PREGUNTA 2 (10 puntos)

Elabore un proyecto denominado "Laboratorio09Preg02", copie en él las clases definidas en la pregunta 1 e incorpórelas al proyecto, luego desarrolle en él las clases que permitan definir y manipular la lista ligada de la figura 1.

Figura 1



La lista debe quedar ordenada de acuerdo con el orden de impresión de los pacientes, primero los de tipo 1, luego 2 y así respectivamente. El orden de impresión no es un atributo de ninguna clase. El proyecto debe ser capaz de leer la información del archivo denominado "AtencionDeCitas.csv" que contiene a todos los pacientes atendidos de acuerdo con las citas realizadas, el archivo es similar al siguiente:

```
A,12484697,Angel Shirakawa Ortega,6720,37.94,A,13633991,Alexander Portugal Raffo,5152,73.85,...
A,20371464,Felicita Villegas Leiva,5152,83.34,U,20688048,Humberto ...
...
```

El archivo está compuesto por varias líneas, contienen los datos de los pacientes que llegan a una institución de salud para ser atendidos. En cada línea puede haber más de un paciente. Los pacientes pueden ser de tres tipos diferentes: de Emergencia, de Urgencia o Ambulatorio.

Los datos de cada paciente estarán dados por 4 datos fijos: DNI, Nombre, tipo de paciente (caracter: 'E', 'U' o 'A') y el código del médico que lo atendió. Además, un paciente ambulatorio tiene un dato adicional, un valor de punto flotante que es el porcentaje de la tarifa de atención que pagará por tener seguro. Un paciente de urgencia tendrá dos datos adicionales, el porcentaje que pagará por seguro y un

porcentaje adicional de descuento por su situación de urgencia. Los pacientes de emergencia deben proporcionar el nombre y número de teléfono de una persona de referencia a la que se pueda acudir en caso se requiera.

El tipo de paciente **no** se colocará en la estructura, debido a que las personas deberán ser manipuladas con polimorfismo. Luego del llenado de la lista se deberá emitir un reporte, con el orden de impresión de acuerdo con el tipo de paciente, generando un archivo de texto similar al que se muestra a continuación:

INSTITUTO DE SALUD SA					
PACIENTES ATENDIDOS DE MANERA AMBULATORIA					
No.	DNI	Paciente	Medico	% por Seg.	
1)	67548237	Maria Carpio Quispe	2251	25.65	
...	
PACIENTES ATENDIDOS POR URGENCIA					
No.	DNI	Paciente	Medico	% por Seg.	% Urgencia
1)	76412542	Pablo Sanchez Gonzales	1289	23.05	10.00
...
PACIENTES ATENDIDOS POR EMERGENCIA					
No.	DNI	Paciente	Medico	Referencia	
1)	35461278	Juan Perez Gomez	3345	Ana Perez	9112253
2)	41627199	Pedro Castro Garcia	6632	Jose Quispe	7812354
...
Atendidos por seguro: 85 pacientes					

En el proyecto "Laboratorio09Preg02", desarrolle la Clase **ListaDePacientes**, esta servirá para ejecutar el proceso de creación de la lista e impresión. Estos métodos reciben como parámetros los nombres de los archivos a trabajar. Considerando el detalle, la función main contendrá el siguiente código:

```
#include "ListaDePacientes.h"
int main(int argc, char** argv) {
    Lpaciente l;
    l.crealista("AtencionDeCitas.csv");
    l.imprimelista("Reporte.txt");
    return 0;
}
```

**No puede
cambiar este
código**

Los métodos que se muestran en el main, deben invocar a las clases y selectores desarrollados anteriormente.

Al finalizar la práctica, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para esta este laboratorio.

Profesores del curso: Miguel Guanira
Rony Cueva

San Miguel, 11 de diciembre del 2020.