

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

7ma 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.
- 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 PROBIDAD 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.

Cuestionario:

En este laboratorio se desarrollarán diversas clases y sobrecargas de operadores, que le permita manejar la atención dada por los médicos que laboran en una institución de salud.

Cree una carpeta denominada "Laboratorio07-2020-2", dentro de ella colocará los proyectos solicitados en las preguntas 1 y 2. **DEBE RESPETAR ESTE REQUERIMIENTO.**

PREGUNTA 1 (15 puntos): Creación de clases

Se solicita que desarrolle un proyecto denominado "LAB07_PREG01_CLASES" en la cual se definirán dos clases que contendrán la información de cada médico y la de los pacientes de la institución. Además, definirá una serie de operadores sobrecargados ligados a las clases que permitirá manejarlas.

Las clases se definen a continuación:

➤ **Para manejar los médicos:** La clase se denominará "**Medico**" y deberá contener lo siguiente (ver adicionalmente el archivo descrito más adelante): un atributo denominado **codigo (char*)**, éste almacena el código del médico, el código estará formado por seis caracteres alfanuméricos, un atributo denominado **nombre** definido por una cadena de caracteres (char*), el atributo denominado **especialidad**, también definido por una cadena de caracteres (char*), el atributo denominado **tarifa** que almacena el monto por consulta que cobra el médico (valor de punto flotante), el atributo **fechas**, definido por un arreglo de enteros (int), que almacenará las fechas en que fueron atendidos o están por ser atendidos los paciente del médico, se almacenarán en el formato **AAAAMMDD**, el atributo denominado **pacientes**, también un arreglo de enteros, que contiene los códigos de cada paciente atendido o por atender, el atributo denominado **prima**, arreglo de valores de punto flotante que almacenará el porcentaje sobre el precio de la consulta que ha pagado o pagará el paciente si tiene un seguro, 100 si no lo tiene, el atributo denominado **estado**: arreglo de tipo caracter que guarda la condición del paciente ('A' si ya fue atendido, 'R' si ha reservado una cita para un día determinado o 'E' si la cita ha sido anulada), el atributo denominado **numPac** que almacena el número de pacientes almacenado en el arreglo, el atributo denominado **ingresosP**, valor de punto flotante que almacenará lo que recibirá el médico por el pago que harán los pacientes y el atributo denominado **ingresosS**, que acumulará el monto que debe solicitar el médico a la compañía de seguros para cubrir el monto de sus consultas. El tamaño de los arreglos se definirá en 100.

➤ **Para manejar los pacientes:** La clase se denominará "**Paciente**", y deberá contener lo siguiente: un atributo denominado **fecha** que contiene un valor entero que con la fecha en que fue atendido el paciente o la fecha de reserva de la cita, la fecha se almacenará en el formato **AAAAMMDD**, el atributo denominado **codigo** que guarda un valor entero con el código del paciente, el atributo denominado **nombre**, cadena de caracteres (char*), con el nombre del paciente, el atributo denominado **prima**, con el porcentaje que paga por consulta (valor de punto flotante), 100 si no tiene seguro y el atributo denominado **estado**: carácter, 'A' si ya fue atendido, 'R' si reservó una cita o 'E' si esta anulada.

"DEBE EMPLEAR OBLIGATORIAMENTE LOS NOMBRES DADOS DE LAS CLASES Y SUS ATRIBUTOS"

"LAS CADENAS DE CARACTERES DEBEN MANEJARSE DE MANERA EXACTA"

Las operaciones que se permitirá realizar a través de sobrecargas de operadores se definen a continuación:

- **Lectura:** Sobrecargando el operador >> de modo que permita leer un médico de un archivo como se muestra más adelante. La operación (**arch >> Medico;**) involucrará un archivo de textos y lo almacena en Medico.
- **Lectura:** Sobrecargando el operador >> de modo que permita leer un paciente de un archivo como se muestra más adelante. La operación (**arch >> Paciente;**) involucrará un archivo de textos y lo almacena en Paciente. La operación no involucrará a la fecha, ni en la lectura ni en la asignación. El operador permitirá una encadenación de varias de estas operaciones.
- **Agregación:** sobrecargando el operador + de modo que permita agregar un paciente a la lista de un médico. La operación (**Medico + Paciente;**) colocará los datos del paciente al final de la lista, modificando el número de pacientes.
- **Modificación de estado:** sobrecargando el operador * de modo que permite pasar del estado 'R' al estado 'A' a todos los pacientes que solicitaron una cita en una fecha determinada. La operación (**Medico * fecha;**) involucrará un objeto Medico con una fecha dada por un valor entero de la forma **AAAAMMDD**.
- **Cancelación de citas:** sobrecargando el operador - de modo que permite cancelar todas las citas de una fecha dada. La operación (**Medico - fecha;**) cambiará al estado 'E' a los datos del arreglo de aquellos pacientes en estado 'R' de una fecha dada por un valor entero de la forma **AAAAMMDD**.

- **Cálculo de ingresos:** sobrecargando el operador ++ de modo que permita calcular el monto que ha recibido el médico directamente de sus pacientes atendidos y el monto que debe solicitar a la compañía de seguros para cubrir el monto de sus consultas.
- **Impresión:** sobrecargando el operador << de modo que permita imprimir la información de un médico de acuerdo al formato descrito más adelante. La operación (**arch << Medico;**) permitirá imprimir en un archivo de textos los datos contenidos en un Medico. El formato será el siguiente:

| | | | | |
|-------------------------------------|------------|----------|--------|----------|
| Médico: Arca/Amezquita/Edric-Ronald | | | | |
| Código: CA0023 | | | | |
| Especialidad: Cardiología | | | | |
| Honorarios por cita: 735.00 | | | | |
| ===== | | | | |
| Pacientes atendidos: | | | | |
| ----- | | | | |
| No. | Fecha | Paciente | Pago | Seguro |
| 1) | 03/06/2020 | 346281 | 230.50 | 504.50 |
| 2) | 03/06/2020 | 173245 | 0.00 | 735.00 |
| ... | ... | ... | ... | ... |
| 53) | 15/09/2020 | 735571 | 10.35 | 724.65 |
| ===== | | | | |
| Pacientes por atender: | | | | |
| ----- | | | | |
| No. | Fecha | Paciente | | |
| 1) | 25/09 | 123452 | | |
| ... | ... | ... | ... | ... |
| ===== | | | | |
| Monto total recibido por pacientes: | | | | 5843.25 |
| Monto a solicitar al seguro: | | | | 33111.75 |

El reporte debe estar perfectamente tabulado (sin usar el carácter '\t').

Consideraciones:

La solución debe contemplar la elaboración de un proyecto que implemente y pruebe las clases y sus sobrecargas. La prueba de las sobrecargas debe ser hecha lo más simple posible pero que muestre claramente los resultados correctos (**No debe solucionar aquí el problema de la pregunta 2**).

PREGUNTA 2 (5 puntos): Prueba final

Desarrolle un proyecto denominado "LAB07_PREG02_CARGA" en el cual se utilizará las clases desarrolladas en la pregunta anterior. Este proyecto solucionará el problema planteado a continuación:

- a) Leer los datos de los médicos contenidos en un archivo de textos como se muestra a continuación y los coloque en un arreglo de la clase Medico:

| | | | |
|--------|----------------------------------|--------------|--------|
| CA0023 | López/Castro/Juan-Carlos-Alfonso | Cardiología | 735.00 |
| DE0105 | Ramírez/Pérez/Ana-Cecilia | Dermatología | 490.0 |
| ... | | | |

En cada línea aparece el código, nombre, especialidad y tarifa de un médico

- b) Leer los datos de los pacientes que ya fueron atendidos o que han solicitado una cita con algún médico. El archivo que contiene estos datos es similar al siguiente:

```
10/09/2020 A 211054 Pedro/López 25.00 LP3465 R 456312 Juan/Chávez 15.5 CJ1052 ...
12/09/2020 R 010704 Valentina/Ruiz 65.00 RV0704 R 100306 Naomi/López 75.5 PL3465...
...
```

En cada línea aparece una fecha seguida de los pacientes que han sido atendidos o se atenderán en la fecha dada, los datos de los pacientes estarán dados por el estado, el código y nombre del paciente, el porcentaje de la tarifa que pagará y el código del médico que lo atendió o lo atenderá. El programa debe colocar los pacientes en las respectivas listas de los médicos.

- c) Leer los datos de las acciones a realizar de un archivo similar al siguiente:

```
1 RV0704 23/09/2020 24/09/2020 ...
2 RV0704 23/09/2020 24/09/2020 ...
3 RV0704 MG0755 AR2607 ...
4 RV0704 MG0755 AR2607 ...
...
```

En cada línea del archivo se tendrán los datos necesarios para realizar una operación determinada, primero aparece la operación que se desea realizar (valor entero entre 1 y 4) seguido de código de un médico y a continuación de los datos requeridos. Las operaciones no están en orden, solo se

muestra así en el ejemplo para su mejor entendimiento, éstas puede aparecer varias veces en el archivo con diferentes datos. Las operaciones serán:

- 1: Modificación de estado, se modificará el estado de los pacientes para varias fechas.
- 2: Cancelación de citas, se cancelarán las citas de los pacientes en las fechas dadas.
- 3: Calculo de ingresos: se harán los cálculos para los médicos indicados.
- 4: Impresión: se imprimirán los reportes para los médicos indicados.

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, 27 de noviembre del 2020.