

PROGRAMACIÓN II - GRUPO A

PRÁCTICA Nº4

Miércoles, 12 de abril 2023

INSTRUCCIONES

1. Sólo se podrá utilizar una hoja con el “chuletero” elaborado por cada alumno con sintaxis del lenguaje. No se permitirá la consulta de ningún otro tipo de material durante la realización de la práctica.
2. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de prácticas de PII del grupo C **antes de la hora de finalización de la sesión de prácticas**.
3. Aunque las prácticas se tengan que realizar en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual, la misma práctica**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en la calificación del estudiante que no la haya entregado**.
4. Se debe entregar **los ficheros** correspondientes al ejercicio en el formato más adecuado (.cpp o .h), **sin comprimir**.
5. Los ficheros entregados deben **incluir, al principio de cada fichero entregado, el nombre de los integrantes del equipo**.
6. El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su **descalificación**.

Criterios generales de evaluación

Funciones/Métodos: Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple.	40%
Tipos de datos y variables: <ul style="list-style-type: none"> • Uso de variables globales (fuera del ámbito de una función). • Si no se usan los tipos contenedor vistos en clase (std::array; std::vector; std::set; std::string, etc.) para las variables que lo necesiten. 	0%
El programa no compila o no se asemeja a lo pedido.	0%
Si no se cumplen los criterios de entrega indicados en la actividad/examen .	0%

Criterios particulares de evaluación

El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegidos son adecuados.	100%

IMPORTANTE:

- Todos los ejercicios del examen deberán ser resueltos de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general** y **no particular** para unos determinados datos/valores.
- Todos los **ejercicios resueltos sin utilizar funciones** cuando sea apropiado se valorarán con una **nota máxima del 60% sobre la calificación prevista**.
- Se recomienda una **primera lectura del enunciado de la práctica** para planificar su realización.
- Conviene ir **probando los diferentes recursos programados** de forma gradual. Programar todo seguido y realizar las pruebas al final cuando quedan 10 minutos para la entrega, suele acabar con **errores de compilación sin resolver en la entrega** y por lo tanto con una calificación de 0 puntos.

Práctica PII_P4GA

Esta práctica consiste en desarrollar en C++11 los elementos adecuados para que el código correspondiente al **programa** y **ficheros cabecera proporcionados** en la actividad funcionen correctamente y cumplan con las especificaciones que se indican.

IMPORTANTE: Se pueden comentar las partes del programa que no se consigan implementar para **evitar errores de compilación**, en cuyo caso, la evaluación de la práctica tendrá su correspondiente **penalización proporcional a la parte que no se haya desarrollado**.

Completar de forma adecuada los ficheros proporcionados para crear la jerarquía de clases que aparecen en la figura 1:

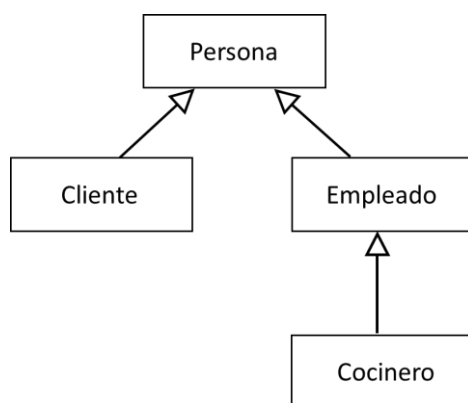


Figura 1. Jerarquía de clases para elementos de un restaurante

El programa que se proporciona presenta un menú que permite al usuario seleccionar la operación que desea realizar. Hay que completar cada opción para que solicite al usuario la información necesaria para llevar a cabo dicha operación. El programa terminará cuando la opción seleccionada del menú sea FIN. Cada opción del menú, siempre que sea adecuado, deberá implementarse a través de una/s función/es y no dejar el código en el main. **(2 puntos)**

1. Solicitar datos para una persona.
2. Mostrar datos de una persona.
3. Solicitar datos para un cliente.
4. Mostrar datos de un cliente.
5. Solicitar datos para un empleado.
6. Mostrar datos de un empleado.
7. Solicitar datos para un cocinero.
8. Mostrar datos de un cocinero.
0. FIN

Para organizar la información se han definido las siguientes clases:

Clase Persona (2 puntos)

- Atributos: nombre, apellidos y DNI.
- Métodos:
 - Constructor por defecto con inicializadores.
 - Constructor con parámetros.
 - Constructor por copia.
 - Métodos **gets** y **sets** para cada atributo.
 - Método **mostrarPerso** que permita mostrar el valor de los atributos de la persona.

Clase Cliente (2 puntos)

- Atributos: descuento (%), fecha reserva (día y mes) y hora de reserva (hora y minutos).
- Métodos:
 - Constructor por defecto con inicializadores.
 - Constructor con parámetros.
 - Constructor por copia.
 - Métodos **gets** y **sets** para cada atributo.
 - Método **mostrarClien** que permita mostrar el valor de los atributos del cliente.

Clase Empleado (2 puntos)

- Atributos: años de antigüedad, salario, categoría y días de vacaciones disponibles.
- Métodos:
 - Constructor por defecto con inicializadores.
 - Constructor con parámetros.
 - Constructor por copia.
 - Métodos **gets** y **sets** para cada atributo.
 - Método **mostrarEmple** que permita mostrar el valor de los atributos del empleado.

Clase Cocinero (2 puntos)

- Atributos: equipo (número de personas a su cargo), tipo de cocina y número de premios.
- Métodos:
 - Constructor por defecto con inicializadores.
 - Constructor con parámetros.
 - Constructor por copia.
 - Métodos **gets** y **sets** para cada atributo.
 - Método **mostrarCocin** que permita mostrar el valor de los atributos del cocinero.

Si se considera necesario, se pueden definir nuevos métodos en cada una de las clases anteriores.