Universitat Oberta de Catalunya uoc.edu

Fundamentos de Programación
PEC5 - 20191

Fecha límite de entrega: 04/11/2019

Estudiante
Apellidos:
Nombre:

Objetivos

- Saber definir correctamente los tipos de datos estructurados.
- Aprender a utilizar la entrada y salida para modificar campos concretos de estructuras de datos sencillos.
- Saber utilizar la as funciones de conversión de tipos cuando sea necesario.

Formato y fecha de entrega

La PEC se ha de entregar antes del día **4 de noviembre de 2019 a las** 23:59.

Para la entrega se deberá entregar un archivo en formato ZIP, que contenga:

- Este documento con la respuesta del ejercicio 1 y el apartado b del ejercicio 2
- Un workspace de Codelite que contenga los archivos .c pedidos al ejercicio 2a

Hay que hacer la entrega en el apartado de entregas de EC del aula de teoría.

Enunciado

Siguiendo con la ayuda que proporcionamos a la compañía UOCBookings, nos han pedido nuestra colaboración para crear un programa que les ayude a gestionar los datos de sus hoteles. En este ejercicio trabajaremos con tipos de datos estructurados juntamente con la entrada y salida interactiva para gestionar los datos de los hoteles.

Nos proporcionan el siguiente esqueleto que deberéis completar

```
type
   tTypeHotel = {BUDGET, INN, RESORT,CONDO, LUXURY, COUNTRY}
end type

algorithm UOCBookings
var

end var

...

end algorithm
```

Ejercicio 1: Tipos estructurados de datos [50%]

Apartado a: [20%] Definir en **lenguaje algorítmico** el tipo de datos estructurado *tHotel,* que representa la información de un hotel. Los datos que se quieren guardar son las siguientes:

- *id.* Identificador del hotel. Se declara cómo un entero.
- *brand.* Cadena hotelera a la que pertenece el hotel. Se declara cómo un string, con una longitud máxima de 15 caracteres, sin espacios en blanco.
- *name.* Nombre del hotel. Se declara cómo un string, con una longitud máxima de 15 caracteres, sin espacios en blanco.
- *hotelType.* Tipo de hotel que puede ser BUDGET, INN, RESORT, CONDO, LUXURY o COUNTRY. El tipo enumerado ya viene declarado en el esqueleto que os proporcionan.
- *city.* Ciudad donde está el hotel. Se declara cómo un string, Se declara cómo un string, con una longitud máxima de 15 caracteres, sin espacios en blanco.
- *category*, El número de estrellas del hotel. Se declara cómo un entero.
- *priceDouble,* Precio de la habitación doble. Se declara cómo un real.

- *distanceFromCityCenter,* Distancia del hotel al centro de la ciudad. Se declara cómo un real.
- *hasPool,* Un booleano que indica si el hotel tiene o no piscina.
- *hasGym,* Un booleano que indica si el hotel tiene o no gimnasio.
- *closeToSubway,* Un booleano que indica si el hotel tiene o no una estación de metro cerca.
- *percentOccupation,* Porcentaje de ocupación del hotel. Se declara como un real.

Apartado b: [60%] Completad el algoritmo que os proporcionan haciendo lo siguiente:

- declarar las variables necesarias para resolver los siguientes puntos.
- leer los datos de tres hoteles del canal estándar de entrada y guardarlos en tres variables del tipo declarado en el apartado a) (*tHotel*).
- leer del canal estándar de entrada el precio máximo que se quiere pagar.
- leer del canal estándar de entrada el nombre de la ciudad dónde se busca el hotel.
- Hacer el siguiente cálculo: Buscar cuál de los tres hoteles es el que está en la ciudad indicada, tiene el mejor precio y el precio no supera el máximo indicado. Si hay dos que cumplen las mismas condiciones, escogemos el primero entrado.
- Mostrar en el canal estándar de salida los datos del hotel seleccionado. Los datos que se deben mostrar son: el nombre del hotel, la ciudad, el precio de la habitación doble, la categoría y el tipo de hotel.
- Si ninguno de los tres hoteles cumple con las condiciones requeridas mostrar un mensaje por el canal estándar de salida indicándolo.

Nota: En lenguaje algorítmico, para leer los datos del tipo *tHotelType*, disponéis de la función *readHotelType().* Y para escribirlos disponéis de la acción *writeHotelType(nombreVariable).*

Apartado c: [20%] Explica cómo se podría enfocar este ejercicio para poder dar de alta de manera interactiva un elevado número de hoteles. No hay que hacer el diseño, simplemente tenéis que razonar la propuesta.

Ejercicio 2: Programación en C [50%]

Apartado a: [80%] Implementad, en **lenguaje C**, el algoritmo del ejercicio anterior. Recordad que en lenguaje C es necesario definir previamente la longitud máxima de las cadenas de caracteres, típicamente mediante una constante.

Apartado b: [20%] Como en las anteriores PEC se pide que indiquéis 4 juegos de prueba que utilizaríais y justifiquéis el por qué los habéis escogido (no hace falta dar todos los datos de los hoteles, solamente los valores de las variables clave para ejecutar y probar el algoritmo).

b1) Ninguno de los hoteles que están en la ciudad tiene un precio lo suficientemente bajo para las condiciones establecidas, para probar que se filtra bien el precio máximo establecido.

Datos de entrada	
Nombre variable	**Valor entrada**

Datos de salida

b2) Todos los hoteles cumplen la condición del precio pero los tres están en otras ciudades, para comprobar que se filtra bien la ciudad.

Datos de entrada	
Nombre variable	Valor entrada

Datos de salida

b3) Hay dos hoteles que cumplen las condiciones y se comprueba que realmente se escoge el de mejor precio.

Datos de entrada	
Nombre variable	Valor entrada

Datos de salida	

B4) Hay dos hoteles que cumplen las condiciones y tienen exactamente el mismo precio. Se comprueba que se muestra el primero

Datos de entrada	
Nombre variable	**Valor entrada**

Datos de salida	

Criterios de corrección:

En el ejercicio 1:

- Que se siga la notación algorítmica utilizada en la asignatura. Véase documento Nomenclator la XWiki de contenido.
- Que se siguen las instrucciones dadas y el algoritmo responda al problema planteado.
- Que se utilice correctamente la estructura alternativa y el tipo de datos estructurado.
- Que se razone correctamente la respuesta del apartado c del ejercicio 1.

En el ejercicio 2:

- Que el programa se adecue a las indicaciones dadas.
- Que el programa compile y funcione de acuerdo con lo que se pide.
- Que se respeten los criterios de estilo de programación C. Véase la Guía de estilo de programación en C que tenéis en la xWiki de contenido.
- Que se declaren los tipos adecuados según el tipo de datos que representa.