

Fundamentos de Programación

PEC1 - 20191

Fecha límite de entrega: 30/09/2019

Estudiante

Apellidos: Delgado Flores

Nombre: Pablo José

Objetivos

- Saber identificar las variables de tipos básicos necesarias para resolver un problema.
- Saber leer y escribir variables de tipos básicos.

Formato y fecha de entrega

La PEC se debe entregar antes del día 30 de septiembre de 2019 a las 23:59.

Se debe entregar un fichero en formato ZIP, que contenga:

- Este mismo documento con la respuesta del ejercicio 1 y el último apartado del ejercicio 2
- Un workspace Codelite que contenga el fichero .c solicitados en el primer apartado del ejercicio 2

La entrega se debe hacer en el apartado de entregas de la EC del aula de teoría.

Enunciado

Este semestre la compañía UOCBookings nos han pedido crear una aplicación para gestionar hoteles y sus clientes.

Para dar respuesta a esta petición, a través de las PEC, iremos creando una pequeña parte de la aplicación, que gestionará los hoteles. El resto, lo completaremos en las prácticas.

Para empezar, nos piden lo siguiente:

Ejercicio 1: Declaración de variables [50 %]

Diseñar en lenguaje algorítmico, un algoritmo que haga lo siguiente:

Apartado a [40%]. Declarar las variables y tipos enumerados necesarios para gestionar los datos de un hotel. Para cada hotel, de momento, se necesita guardar la siguiente información:

- Un identificador de tipo entero, que es el número que identifica el hotel.
- Un enumerado que indique el tipo de hotel. Puede tomar los valores {BUDGET, INN, RESORT, CONDO, LUXURY, COUNTRY}.
- Un entero que indique la categoría del hotel (número de estrellas).
- Un carácter que indique una valoración del hotel interna de UOCBookings.
- Un entero que indique el número de habitaciones.
- Un real que indique el precio de la habitación doble.
- Un real que indique la distancia del hotel al centro de la ciudad.
- Un booleano que indique si tiene piscina.
- Un real que indique el porcentaje de ocupación del hotel.

Apartado b [30%]. Pedir y leer, por el canal estándar de entrada, los valores de las variables de tipo entero, carácter y real declaradas en el apartado anterior de un hotel en concreto.

Para la lectura se indicará por el canal estándar de salida qué información se espera que el usuario introduzca. Por ejemplo, para leer una variable *age* correspondiente a la edad de una persona, habría que escribir en pseudocódigo:

```
writeString("Enter the current age:");  
age=readInteger();
```

Apartado c [30%]. Mostrar por el canal estándar de salida el valor de las variables leídas del hotel, indicando qué es cada información. Siguiendo con el ejemplo haríamos:

```
writeString("Age:");  
writeInteger(age);
```

SOLUCIÓN

{ Lo primero será declarar el tipo de enumerado correspondiente al tipo de hotel }

type

tHotelT = { BUDGET, INN, RESORT, CONDO, LUXURY, COUNTRY };

end type

algorithm *Hotel-UOC*

var

idHotel: integer;
hotelType: *tHotelT*;
hotelCat: integer;
hotelRating: char;
roomsNumber: integer;
priceDRoom: real;
centerDist: real;
hasPool: boolean;
occupationPercent: real;

end var

writeString("Introduce el ID del hotel: ");
idHotel:=**readInteger**();
writeString("Introduce el número de estrellas del hotel: ");
hotelCat:=**readInteger**();
writeString("Introduce la valoración de UOCBookings: ");
hotelRating:=**readChar**();
writeString("Introduce el número de habitaciones: ");
roomsNumber:=**readInteger**();
writeString("Introduce el precio de una habitación doble: ");
priceDRoom:=**readReal**();
writeString("Introduce la distancia que hay hasta el centro: ");
centerDist:=**readReal**();
writeString("Introduce el porcentaje de ocupación del hotel: ");
occupationPercent:=**readReal**();

writeString("ID del hotel: ");
writeInteger(*idHotel*);
writeString("Categoría del hotel (en número de estrellas: ");
writeInteger(*hotelCat*);
writeString("Valoración interna del hotel en UOCBookings: ");
writeChar(*hotelRating*);

```
writeString("Número de habitaciones: ");
writeInteger(roomsNumber);
writeString("Precio de la habitación doble: ");
writeReal(priceDRoom);
writeString("Distancia hasta el centro: ");
writeReal(centerDist);
writeString("Porcentaje de ocupación del hotel: ");
writeReal(occupationPercent);
```

end algorithm

Ejercicio 2: programación en C [50%]

Apartado a [80%] Codificación

Codificar en C el algoritmo del ejercicio 1. En la programación los reales se deben escribir con dos decimales.

Apartado b [20%] Pruebas / Ejecución del algoritmo

Los algoritmos codificados deben siempre probarse. Es decir, se deben ejecutar los programas dando diferentes valores a las variables de entrada y comprobando que la salida corresponde a los valores esperados. A este proceso se le llama realizar *Juegos de pruebas*. Los juegos de prueba tienen que ser diferentes para comprobar las diferentes situaciones que puede ejecutar el programa

En este apartado se solicita que se diseñe un juego de pruebas. Es decir, que completéis la tabla siguiente indicando para unos valores de entrada concretos, qué salida se espera de la ejecución del programa. Después comprobad que efectivamente el programa hace lo esperado. Se debe justificar porque se ha escogido el juego de pruebas presentado.

SOLUCIÓN

Para la realización del juego de pruebas, he pasado mi código por ACME, el cual me ha ofrecido una serie de datos de entrada que detallo a continuación:

Datos de entrada	
Nombre variable	Valor entrada
idHotel	12
hotelCat	3
hotelRating	G
roomsNumber	246
priceDRoom	80
centerDistance	3.45
occupationPercent	78.91

Datos de salida
12
3
G
246
80.00
3.45
78.91

A pesar de haberse ejecutado correctamente, ACME ha arrojado un error ya que no ha obtenido las mismas cadenas de texto que preveía (esto no quiere decir que esté mal, simplemente no era lo que el programa esperaba). A continuación, he realizado otro juego de pruebas aplicando mis propios datos, para comprobar que los decimales funcionaban correctamente. Esto es lo que he obtenido:

Datos de entrada	
Nombre variable	Valor entrada
idHotel	141
hotelCat	4
hotelRating	A
roomsNumber	100
priceDRoom	27.99
centerDistance	3.487
occupationPercent	77.754

Datos de salida
141
4
A
100
27.99
3.49
77.75

Criterios de corrección:

En el ejercicio 1:

- Que se siga la notación algorítmica utilizada en la asignatura. Ved el documento *Nomenclator* en la xWiki.
- Que se sigan las instrucciones dadas y el algoritmo responda al problema planteado.

En el ejercicio 2:

- Que el programa se adecue a las instrucciones dadas.
- Que el programa compile y que funcione según lo solicitado.
- En la codificación en C del algoritmo, que se declaren los tipos adecuados según el tipo de datos que representa.
- Que se respeten los criterios de estilo de programación C. Ved la *Guía de estilo de programación en C* que tenéis en la Wiki.