

PROGRAMACIÓN I

T.U.P.

U.T.N. F.R.P.

1

Programación I

Estrategia

2

Repasemos :

Iterativos :

Para que sirven ???

Cuales diferenciamos???

Que uso se le da a cada uno ????

PARA vs Mientras :
Como diferenciamos los
mecanismos de control de bucle?
Cuando uso cada una?

Repetir vs Mientras :
Como diferenciamos los
momentos de control de bucle?
Cuando uso cada una?

2

Hasta aquí....

3

- Hemos comenzado a programar en c#.
- Planteamos que C# es un lenguaje orientado a objetos .
- Pero que significa esa orientación.

3

Ejemplo 1

Una empresa de transporte cobra a sus pasajeros en función de la distancia recorrida según el siguiente cuadro:

Tarifa 1 de \$10 se aplica desde el inicio del viaje e incluye los primeros 2 Km.

Tarifa 2 de \$3.2 se aplica por kilómetro excedente a la tarifa 1 hasta los 60 Km.

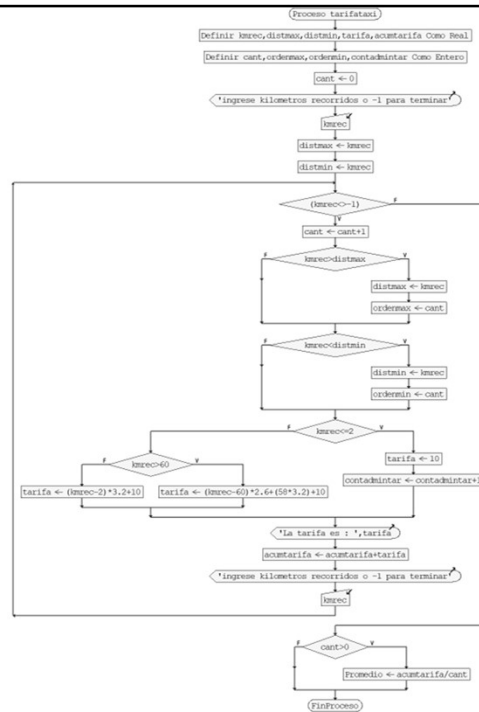
Tarifa 3 de \$2.6 se aplica por kilómetro excedente a la tarifa 2 a partir de los 61Km.

Desarrolle una aplicación que informe el importe a cobrar al pasajero independientemente de las consideraciones anteriores y para cualquier kilometraje.

La aplicación debe aceptar valores de kilometraje recorridos, y finalizar cuando se ingrese -1 . Al finalizar debe informar la cantidad total de valores ingresados , el valor promedio, el valor kilómetros recorridos en el viaje mas largo, la cantidad de viajes con tarifa mínima registrados, el numero de orden en que ingreso viaje con menos kilómetros cargado y la distancia recorrida.

4

Ejemplo 1



5

Ejemplo 1

```

using System;

namespace tarifa {
    class tarifataxi {

        static void Main(string[] args)
        {
            double acmtarifa;
            int cant, contadmintar, ordenmax, ordenmin;
            double distmax, distmin, kmrec, promedio, tarifa;
            cant = 0;
            Console.WriteLine("ingrese kilometros recorridos o -1 para
terminar");

            kmrec = Convert.ToDouble(Console.ReadLine());
            distmax = kmrec;
            distmin = kmrec;

```

6

Ejemplo 1

```

while ((kmrec!=1)) {
    cant = cant+1;
    if (kmrec>distmax)
        { distmax = kmrec;
          ordenmax = cant;
        }
    if (kmrec<distmin)
        { distmin = kmrec;
          ordenmin = cant;
        }
    if (kmrec<=2)
        { tarifa = 10;
          contadmintar = contadmintar+1;
        }
    else {
        if (kmrec>60) {
            tarifa = (kmrec-60)*2.6+(58*3.2)+10;
        }
        else {
            { tarifa = (kmrec-2)*3.2+10;
            }
        }
    }
    Console.WriteLine("La tarifa es : "+tarifa);
    acumtarifa = acumtarifa+tarifa;
    Console.WriteLine("Ingrese kilometros recorridos o -1 para terminar");
    kmrec = Convert.ToDouble(Console.ReadLine());
}

```

7

Ejemplo 1

```

        kmrec = Convert.ToDouble(Console.ReadLine());
    }
    if (cant>0) {
        promedio = acumtarifa/cant;
    }
    /// El resto de la información a mostrar la completa cada alumno !!!
}
}
}
}

```

8

Paradigmas

¿Qué es un paradigma de programación?

Un paradigma de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas (con la solución de múltiples "problemas" se construye una aplicación).

Los lenguajes de programación son basados en uno o más paradigmas. Por ejemplo: Smalltalk y Java son lenguajes basados en el paradigma orientado a objetos. El lenguaje de programación Scheme, en cambio, soporta sólo programación funcional.

En cambio Python, soporta múltiples paradigmas. prototipos, paradigma orientado a aspectos, etc.

9

Paradigmas

PARADIGMA DE PROGRAMACIÓN

oParadigma de programación es una propuesta tecnológica adoptada por una comunidad de programadores cuyo núcleo central es incuestionable en cuanto a que unívocamente trata de resolver uno o varios problemas claramente delimitados.

La resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de Software.

oLos lenguajes de programación suelen implementar, a menudo de forma parcial, varios paradigmas

10

Paradigmas	
Paradigmas de Programación	Estructurada
	Funcional
	Orientada a Objetos
	Lógica
	Basada en eventos
	Reactivo

11

Paradigmas
<p>Paradigma Imperativo: describe la programación como una secuencia instrucciones o comandos que cambian el estado de un programa. El código máquina en general está basado en el paradigma imperativo. Su contrario es el paradigma declarativo. En este paradigma es un tipo general que incluye los paradigmas procedimental (procedural), estructurado, y orientado a objetos, entre otros.</p>

12

Paradigmas

Paradigma Declarativo: No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado para encontrar esa solución. Es más complicado de implementar que el paradigma imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados problemas. En este paradigma es un tipo general que incluye los paradigmas Funcional y lógico

13

Paradigmas

Paradigma Estructurado: la programación se divide en bloques (procedimientos y funciones) que pueden o no comunicarse entre sí. Además la programación se controla con secuencia, selección e iteración. Permite reutilizar código programado y otorga una mejor comprensión de la programación. Es contrario al paradigma inestructurado, de poco uso, que no tiene ninguna estructura, es simplemente un “bloque”, como por ejemplo, los archivos batch (.bat).

Paradigma Orientado a Objetos: está basado en la idea de encapsular estado y operaciones en objetos. En general, la programación se resuelve comunicando dichos objetos a través de mensajes (programación orientada a mensajes). Se puede incluir -aunque no formalmente- dentro de este paradigma, el paradigma basado en objetos, que además posee herencia y subtipos entre objetos. Ej.: Simula, Smalltalk, C++, Java, Visual Basic .NET, etc. Su principal ventaja es la reutilización de códigos y su facilidad para pensar soluciones a determinados problemas.

14

Paradigmas

Paradigma Funcional: este paradigma concibe a la computación como la evaluación de funciones matemáticas y evita declarar y cambiar datos. En otras palabras, hace hincapié en la aplicación de las funciones y composición entre ellas, más que en los cambios de estados y la ejecución secuencial de comandos (como lo hace el paradigma procedimental). Permite resolver ciertos problemas de forma elegante y los lenguajes puramente funcionales evitan los efectos secundarios comunes en otro tipo de programaciones.

Paradigma lógico: se basa en la definición de reglas lógicas para luego, a través de un motor de inferencias lógicas, responder preguntas planteadas al sistema y así resolver los problemas. Ej.: prolog.

Otros paradigmas y subparadigmas son: paradigma orientado al sujeto, paradigma reflectante, programación basada en reglas, paradigma basado en restricciones, programación basada en prototipos, paradigma orientado a aspectos, etc.

15

Tipos generales

Tipos de Paradigmas

Imperativos

Estructurado

Orientada a
Objetos

Declarativos

Logica

Funcional

16

Paradigma Funcional

El paradigma funcional considera al **programa como una función matemática**, donde el dominio representaría el conjunto de todas las entradas posibles (inputs) y el rango sería el conjunto de todas las salidas posibles (outputs).

Ventajas:

Más fáciles de escribir y depurar. Uso de **arquitecturas paralelas**.

Desventajas:

Carecen de librerías, interfaces con otros lenguajes y herramientas de depuración.-Al no utilizar sentencias, no existen asignaciones, por lo que una vez que **las variables** asumen un valor, **no cambian durante la ejecución**. **Modelo alejado de Von Neumann**

Lenguajes de programación:

*Haskell Scala Miranda Scheme Lisp SAP Standard

17

Paradigma de Programacion Estructurada

La programación estructurada es una forma de escribir programas de computadora utilizando ciertas instrucciones de control (bucles y condicionales), se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución, es decir, **se emplea un algoritmo que describe los pasos necesarios para solucionar el problema**.

VENTAJAS

La programación estructurada se caracteriza por tener **lenguajes sencillos** de comprender para una persona con basto conocimiento ya que se puede leer de secuencia por ello se considera semi-natural la lógica que se emplea en este tipo de programación es mas visible lo que facilita las **pruebas de escritorio** y la corrección de errores.

DESVENTAJAS

El principal inconveniente de este método de programación, es que se obtiene **un único bloque** de programa, **que cuando se hace demasiado grande puede resultar problemático su manejo**

LENGUAJES

ALGOL PL/I Ada BASIC C Fortran Pascal

18

Paradigma de Programacion Estructurada

La programación estructurada es una forma de escribir programas de computadora de forma clara, utilizando únicamente tres estructuras: secuencia, selección e iteración; siendo innecesario y no permitiéndose el uso de la instrucción o instrucciones de transferencia incondicional.

La programación estructurada surge a finales de los años 1960 con el objetivo de realizar programas confiables y eficientes, y que además fueran escritos de manera de facilitar su comprensión posterior.

Hoy en día las aplicaciones informáticas son mucho más ambiciosas que las necesidades de aquellos años, por lo que se desarrollaron nuevas técnicas, tales como la programación orientada a objetos y el desarrollo de entornos de programación que facilitan la programación de grandes aplicaciones.

De todas formas, el paradigma estructurado tiene vigencia en muchos ámbitos de desarrollo de programas y constituye una buena forma de iniciarse en la programación de computadoras.

19

Teorema del programa estructurado

El teorema del programa estructurado, de *Böhm-Jacopini*, demuestra que todo programa puede escribirse utilizando únicamente las tres instrucciones de control siguientes:

Secuencia

Instrucción condicional.

Iteración (bucle de instrucciones) con condición al principio.

Solamente con estas tres estructuras o "patrones lógicos" se pueden escribir todos los programas y aplicaciones posibles. Si bien los lenguajes de programación tienen un mayor repertorio de estructuras de control, éstas pueden ser construidas mediante las tres básicas.

20

Paradigma POO

PROGRAMACIÓN ORIENTADA A OBJETOS

El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

Ventajas:

La **facilidad de re-utilización de código en diferentes proyectos**.

La **facilidad de añadir, suprimir o modificar nuevos objetos** nos permite hacer modificaciones de una forma muy sencilla.

Debido a las sencillez para abstraer el problema, los programas orientados a objetos son más **sencillos de leer y comprender**

Desventajas

Complejidad para adaptarse

La **necesidad de utilizar bibliotecas** de clases obliga a su aprendizaje y entrenamiento.

Lenguajes de programación:

SIMULA SMALLTALK ACTOR Eiffel JAVA C++ C# PYTHON

21

Paradigma Orientado a Eventos

PARADIGMA ORIENTADO A EVENTOS

La programación dirigida por eventos es un paradigma de la programación en el que tanto **la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema**, definidos por el usuario o que ellos mismos provoquen.

VENTAJAS

Uno de los gran enfoques de la programación orientada a eventos es la **calidad en la interfaz grafica de usuario** la cual es importante para que el administrador de el evento le facilite el uso del software.

DESVENTAJAS

La mayoría de herramientas de creación de interfaces gráficas de usuario se distribuyen como frameworks.

Los **frameworks** implementan el bucle de eventos y la cola de eventos para que no tengamos que implementarlos nosotros mismos. pero el mecanismo queda encerrado dentro del framework lo cual **dificulta su programación y ejecución**.

LENGUAJES DE PROGRAMACIÓN

Visual Basic Java script Visual C++ Visual C# Lexico

22

Procedimientos y funciones

La programación estructurada es una forma de escribir programas de computadora utilizando ciertas instrucciones de control (bucles y condicionales), se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución, es decir, **se emplea un algoritmo que describe los pasos necesarios para solucionar el problema.**

Dentro de la programación estructura se utilizan también funciones y procedimientos como formas de agrupar ciertos fragmentos de código , que se comportan como sub-procesos, que resuelven partes del problema.

De esta manera se incorpora la idea de reutilización de código, en la que se separan distintas partes del programa .

Permite la división del trabajo en el equipo de programación .

La agrupación de funciones por especialidad en bibliotecas .

23

Muchas
Gracias !!!!

24