

UNIVERSIDAD POLITECNICA DE TECAMAC



TAREAS PROGRAMACION VISUAL

PARA OBTENER EL TITULO DE

INGENIERIA EN SOFTWARE

PRESENTA

FLORES MORALES ARIATNA JANETTE

ROMERO PADILLA FRANCISCO OMAR

MATERIA:

PROGRAMACION VISUAL

ASESOR ACADEMICO

MTRO. EMMANUEL TORRES

GRUPO:

5322IS

INTRODUCCION

El diseño de aplicaciones ha cambiado drásticamente, en los últimos años y consecuentemente los estilos de programación, de una programación secuencial, se pasó de una programación modular, para llegar a una programación estructurada con muy buenos fundamentos. A partir de la programación estructurada los esfuerzos se centraron en potenciar el modularidad y la reutilización de código, lo que condujo a la programación orientada a objetos. Pero que los estilos hayan cambiado, no significa que el ultimo elimine a los anteriores, por el contrario, se apoyan en ellos.

1.CONCEPTOS DE PROGRAMACION ORIENTADO A OBJETOS

La programación Orientada a objetos es una técnica particular de programar, que lleva a código las cosas de la vida real. El paradigma más popular es el que se refiere a la Programación Orientada a Objetos o POO. En este paradigma, los programas se modelan en torno a objetos que aglutinan toda la funcionalidad relacionada con ellos. Por eso me he animado a grabar el vídeo que tienes al principio, y a escribir este artículo, en los que te voy a explicar con las palabras más sencillas posibles los principales conceptos de la Programación Orientada a Objetos, independientemente del lenguaje de programación que utilices.

Define de manera genérica cómo van a ser los objetos de determinado tipo.

Encapsulación

Es decir, que todos los datos referentes a un objeto queden «encerrados» dentro de éste y sólo se puede acceder a ellos a través de los miembros que la clase proporcione. Por ejemplo, en el caso de las personas que estábamos viendo, toda la información sobre éstas está circunscrito al ámbito de dicha persona. Así, internamente tenemos un dato que es el nombre de la persona y accedemos a él a través de la propiedad pública Nombre que define la clase que representa a las personas.

Abstracción

Como la propia palabra indica, el principio de abstracción lo que implica es que la clase debe representar las características de la entidad hacia el mundo exterior, pero ocultando la complejidad que llevan aparejada. Así, una clase debe exponer para su uso solo lo que sea necesario. Cómo se haga «por dentro» es irrelevante para los programas que hagan uso de los objetos de esa clase.

Herencia

Del mismo modo en POO cuando una clase hereda de otra obtiene todos los rasgos que tuviese la primera. Dado que una clase es un patrón que define cómo es y cómo se comporta una cierta entidad, una clase que hereda de otra obtiene todos los rasgos de la primera y añade otros nuevos y además también puede modificar algunos de los que ha heredado. Así, en nuestro juego que involucra personas, podemos tener clases de personas más especializadas para representar personajes especiales del juego.

Por ejemplo, podríamos definir clases como Pirata, Piloto o Estratega que heredan de la clase Persona. Por ejemplo, los objetos de la clase Pirata tienen un método nuevo que es Abordar que en el juego sirve para asaltar un barco enemigo. Lo bueno de la herencia es que podemos reutilizar todo lo que tuviésemos en la clase base.

Polimorfismo

En POO, el concepto de polimorfismo se refiere al hecho de que varios objetos de diferentes clases, pero con una base común, se pueden usar de manera indistinta, sin tener que saber de qué clase exacta son para poder hacerlo. La idea es que puedas tratarlos a todos como personas, independientemente del tipo específico de persona que sean y simplemente decirles que hablen. El polimorfismo nos permite utilizar a los objetos de manera genérica, aunque internamente se comporten según su variedad específica.

Por su parte los programas orientados a eventos son los programas típicos de Windows, tales como Netscape, Word, Excel, PowerPoint y otros. Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de las acciones del usuario, que en este caso son llamadas eventos.

El usuario dice si quiere abrir y modificar un fichero existente, o bien comenzar a crear un fichero desde el principio. Estos programas pasan la mayor parte de su tiempo esperando las acciones del usuario (eventos) y respondiendo a ellas. Las acciones que el usuario puede realizar en un momento determinado son variadísimas, y exigen un tipo especial de programación: la programación orientada a eventos.

Este tipo de programación es sensiblemente más complicada que la secuencial y la interactiva, pero con los lenguajes visuales de hoy, se hace sencilla y agradable.

2. CARACTERISTICAS Y APLICACIONES DE EVENTOS

Dependencia de eventos

El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa, contrastando así con el procesamiento por lotes.

Orientada al servicio

Se utiliza para escribir programas diseñados para el servicio sin ralentizar la computadora, ya que la orientación al servicio solo consume poco poder de procesamiento. Además, los servicios se ejecutan por lo general en el trasfondo del sistema operativo.

Eventos

Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema. Cada evento es diferente por naturaleza, algunos requieren que el programa recobre y muestre cierta información, y otros que se inicien algunos cálculos y cambios de estado.

Controlador de eventos

Es una determinada unidad del programa que se activa para reaccionar ante un evento. Es decir, es un tipo de función o método que ejecuta una acción específica cuando se activa un evento determinado.

Funciones de activación

Son funciones que deciden qué código ejecutar cuando se produce un evento específico. Se utilizan para seleccionar qué controlador de eventos emplear al producirse un evento.

Tiempo controlado

Es un código específico que se ejecuta en un momento determinado. Esto significa que es una tarea preestablecida por hacer.

3.CARACTERISTICAS DE COMPONENTES, METODOS VISUALES Y NO VISUALES

Este estilo de programación presenta las siguientes características:

- Visual: los elementos se arrastran y sueltan en el flujo del programa para integrarlos.
- Orientado a acontecimientos: cada paso del programa comienza cuando ocurre un acontecimiento previamente definido.
- Imperativo: la programación consiste en una secuencia de comandos.
- Orientado a objetos: hay objetos individuales con tareas definidas asignadas.

Los programadores suelen utilizar elementos de programación ya definidos, según el lenguaje de programación visual que utilicen. Si los elementos no encajan lógicamente, el editor lo detecta. Estos pueden registrar textos e imágenes, organizarlos de forma lógica y modificar su apariencia o su expresión.

Los componentes no visuales se pueden colocar en los formularios de la misma manera que los controles, aunque en este caso su posición es irrelevante

Componentes interactivos: permiten que el usuario final los manipule, ya sea introduciendo datos, seleccionando elementos, etc. De forma que estos componentes pueden recibir el foco así como los eventos propios del teclado y del ratón.

Componentes gráficos: el propio componente es el encargado de dibujar en la pantalla lo que crea oportuno, bien a través de las funciones básicas del API de Windows (con el objeto TCanvas) o bien a través de otras librerías gráficas, como OpenGL, DirectX, etc. Estos componentes, no suelen recibir eventos del usuario final, aunque si eventos del propio programador, ya que su cometido no suele ir más allá de mostrar ciertos gráficos o imágenes en la pantalla.

La herencia nos permite reutilizar código, haciendo que una clase dada adquirida a todos los atributos y métodos públicos y protegidos de otra, llamada clase padre. Los componentes no-visuales deben heredar directamente de TComponent, ya que este proporciona las características básicas. La peculiaridad es que esta clase no contiene ningún método ni propiedad en la sección Publisher, por lo que puedes ser manejada desde el entorno de Delphi. En realidad, todo el trabajo de esta clase se

limita a publicar las propiedades y eventos que han sido programados en la clase TCustomTreeVieww.

Ahora que ya sabemos quién va a ser el padre de nuestro componente, tenemos que decir qué propiedades y eventos vamos a proporcionar a nuestros usuarios del componente. Puede ser la carpeta raíz de una unidad de disco para mostrar los directorios de toda la unidad, o bien una carpeta cualquiera, para mostrar las subcarpetas de esta.

4. PROCESOS DE DESARROLLO VISUAL EN PROYECTOS DISTRIBUIDOS Y DE ESCRITORIO.

Planifica el horario con un diagrama de Gantt:

Los diagramas de Gantt son muy populares en el mundo de la gestión de proyectos. Una buena forma de plasmar esta información es creando un diagrama de Gantt.

Estructura tu equipo con un organigrama:

Organizar y estructurar tu equipo es vital para el éxito del proyecto. Esta herramienta representa con precisión la estructura del equipo al tiempo que describe la relación entre los distintos miembros del equipo. Esto es especialmente importante para los consultores o autónomos ya que ayuda al cliente a comprender quién es responsable de cada tarea en el equipo.

Visualización de datos

Esta última es siempre la mejor opción, porque la visualización de datos facilita que las personas lo comprendan y lo digieran mejor.

Evaluación de los riesgos con una estructura de desglose

Desde estimaciones de costes bajos y retrasos en el cronograma hasta condiciones fluctuantes del mercado, tu proyecto se puede enfrentar a varios tipos de riesgos en el camino. Puedes utilizar una estructura de desglose de riesgos para ilustrar esto. Una vez que hayas identificado los tipos de riesgos, puedes organizarlos según su nivel y evaluar individualmente cada uno de ellos para desarrollar un plan de acción de respaldo.

Visual Studio proporciona una variedad de plantillas de proyecto para compilar aplicaciones para UWP con C# o C++. Para usar estas plantillas de proyecto, debe incluir la carga de trabajo Desarrollo con la Plataforma universal de Windows al instalar Visual Studio. En el caso de las plantillas de proyecto en C++, también debe incluir el componente opcional Herramientas de Plataformas universales de Windows de C++ (v142) para la carga de trabajo Desarrollo con la Plataforma universal de Windows.

5. REQUERIMIENTOS VISUALES DE PROYECTOS DISTRIBUIDOS Y DE ESCRITORIO.

Es precisamente esta transmisión de conocimientos del cliente a la empresa desarrolladora lo que se conoce cómo toma de requerimientos de un proyecto.

La toma de requerimientos de un proyecto consiste en transmitir el sentido del proyecto a la empresa o persona que lo desarrollará.

En esta etapa se comunica, por ejemplo, en la creación de un sitio web, información importante como:

- El nombre de dominio deseado
- La imagen, marca, logo o colores
- Las páginas que desean que incluya el sitio web (Productos, Quienes somos, Contacto, Agenda una llamada, etc.)
- Información de la empresa para la página de Quienes somos
- Ubicación de la empresa
- Información de contacto
- Productos a vender en línea

La cantidad de requerimientos de un proyecto de software puede ser muy grande y difícil de controlar. Además, se requiere una alta participación del usuario en el proceso de identificar y validar los requerimientos, para garantizar que sean los correctos y cubran sus necesidades. Es una participación que muchas veces no se tiene.

6.HERRAMIENTAS Y LENGUAJES DE PROGRAMACION VISUAL

Entorno de la programación visual

La programación visual permite diseñar y desarrollar aplicaciones con un entorno visual amigable y de fácil uso para el usuario.

El entorno visual comprende la gran variedad de elementos gráficos como ventanas, botones, iconos, entre otros, que sirven para el diseño de aplicaciones y que se relacionan entre ellos para proporcionar funcionalidad a los programas.

Existen varios lenguajes de programación de entorno visual, entre los cuales se citan Visual Basic, Visual C++, Visual C#, Delphi entre otros.

Entorno de la programación orientada a objetos

La programación orientada a objetos consiste en desarrollar programas como un conjunto de objetos, que colaboran entre ellos para realizar tareas.

Elementos de la Programación Visual

Controles

Un control en programación visual es esencialmente un componente con una representación visual.

Eventos

La mayoría de eventos son puestos en acción por el usuario cuando manipula un control en tiempo de ejecución. En una aplicación de formularios Windows Forms, los eventos son generados por controles como botones, cuadros de lista, cuadros de verificación entre otros.

CONCLUSION

La programación se puede ver y escuchar como algo muy complicado de hacer, pero realmente cualquiera lo puede hacer si así se le propone ya que solo se aplica un conocimiento general de las cosas; una vez que se comienza a hacer es muy entretenido y cautivador el ser capaz de crear algo con el solo hecho de pensarlo para luego llevarlo a cabo. Al mismo tiempo vamos aprendiendo a mejorar ese pensamiento, también a poder transmitirlo a una computadora y hacia las demás personas que aún desconocen el gran mundo de la programación.

En conclusión, puedo decir que la forma de programar, ha evolucionado con el paso del tiempo, en la actualidad, existen diversos programas los cuales nos facilitan nuestras actividades que realizamos en nuestra vida diaria, es un gran trabajo el que hacen los programadores, es importante mencionar que no cualquiera puede hacer un programa, se debe tener una preparación para realizar estos.