



Análisis de la programación visual



Materia: Programación Visual

Profesor: Torres Servin Emmanuel

Alumno: RAMIREZ PEREZ CHRISTIAN URIEL

ERICK ALEJANDRO LOPEZ PACHECO

Matricula: 1321124315
1321124248

Grupo: 4322IS

Carrera: Ingeniería en Software

Conceptos de programación orientada a objetos.

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Python es un lenguaje orientado a objetos.

Características y aplicaciones de eventos.

Características

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.

Los eventos en sí pueden ser desde aceptar o rechazar una solicitud de préstamo, denominado evento de alto nivel, hasta que un usuario presione una tecla, esto es un evento de bajo nivel.

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.

Los eventos incluyen al mouse, al teclado, una interfaz de usuario y las acciones que se deben activar en el programa cuando ocurran. Esto significa que el usuario debe interactuar con un objeto en el programa, como hacer clic en un botón del mouse, usar el teclado para seleccionar un botón, etc.

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.

Por ejemplo, podría ser un botón que cuando el usuario haga clic en él muestre un mensaje y cuando vuelva a hacer clic en ese botón cierre el mensaje.

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.

La actualización de Windows es un ejemplo de tiempo controlado, donde el usuario puede establecer cuándo actualizar o cuándo verificar y descargar la actualización.

Aplicaciones

La **programación** dirigida por **eventos** es lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

El evento permite al usuario realizar una serie de acciones lógicas para un programa. sobre un determinado componente, elemento que presta un servicio de comunicación cuando se diseñan interfaces, se da inicio a un conjunto de acciones programadas por el usuario para ese evento concreto.

Los eventos incluyen al mouse, al teclado, una interfaz de usuario y las acciones que se deben activar en el programa cuando ocurran, esto quiere decir que el usuario debe interactuar con un objeto en el programa, como hacer clic en un botón del mouse, usar el teclado para seleccionar un botón etc.

Características de componentes y métodos visuales y no visuales.

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporizadores, cuadros de diálogo -no visibles en la fase de diseño-, etc).

Todos los componentes visuales tienen un método llamado *Show* para mostrarlos y otro llamado *Hide* para ocultarlos.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Un sistema distribuido es un sistema de software cuyos componentes están separados físicamente y conectados entre sí por una red de computadoras. Dichos componentes interactúan entre ellos para lograr una meta común.

Las tres características principales de un sistema distribuido son:

1. Concurrencia de componentes: Los componentes pueden ejecutar sus acciones de manera concurrente e independiente.
2. No hay un reloj global: Los componentes (nodos) de un sistema distribuido no dependen de un reloj que sincronice o indique las acciones de los distintos nodos.
3. Falla independiente de componentes: La falla de un componente no afecta al resto de los componentes.

ordenadores personales. A diferencia de las aplicaciones web, se instalan directamente sobre el sistema operativo.

La portabilidad de las aplicaciones de escritorio se consigue:

- Con múltiples compiladores.
- Con Lenguajes basados en máquina virtual.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Maqueta de interfaz de usuario: Una versión mas detallada y grafica del wireframe, no solo ayuda a hacerse una idea de como funciona su aplicación, también del aspecto

Herramientas y lenguajes de programación visual.

SCRATCH

El lenguaje de programación visual más popular y utilizado es [Scratch](#). Fue lanzado por primera vez en 2007 y tiene como propósito enseñar a los niños a programar **de manera sencilla**.

BLOCKLY

[Blockly](#) es una llamada biblioteca. Proporciona un **editor de programación visual** al que se añaden aplicaciones Android, iOS y web. Blockly también utiliza bloques gráficos que encajan entre ellos

NEPO

NEPO es gratuito, está basado en Scratch y utiliza la biblioteca Blockly. Esta biblioteca se ha ampliado **con funcionalidades propias**. Una ventaja de NEPO son sus interfaces abiertas, que permiten controlar otros sistemas de hardware o robots.

GRAPE

Grape es un entorno de desarrollo gráfico. Permite incluso a los **principiantes en programación** programar con microcontroladores en pasos simples.

APP INVENTOR

App Inventor proviene originalmente de Google. Esta **interfaz gráfica** permite programar aplicaciones para teléfonos móviles Android con bloques gráficos.

ARDUBLOCK

Este lenguaje de programación gráfica está especialmente diseñado para programar el **microcontrolador Arduino** sin introducir texto.

PURE DATA

Este lenguaje de programación visual está orientado tanto a flujos de datos como a los entornos de desarrollo. Pure Data permite producir **software multimedia interactivo**, por ejemplo, para sintetizadores.

LEGO MINDSTORM

Una serie de productos del fabricante de juguetes Lego, cuyo núcleo es la **pieza de Lego programable**: los motores eléctricos, sensores y piezas de tecnología propios de Lego permiten construir y programar robots y otros sistemas interactivos.