

Събитийно проектиране. Основни понятия - Събитие; Обработка; Източник; Методи

I. Събитийно проектиране В традиционния процедурен модел на програмиране приложението се управлява от определена последователност от команди. Именно в ясно зададената последователност се изпълняват разработените части на програмния код. Изпълнението на приложението започва от първия ред на програмата и следва пътя в строго зададена последователност, която се изпълнява или линейно, или разклонено, като при необходимост се извикват различните разработени части код или процедури. Смесът на събитийно-управляемото програмиране е, че вместо подробното описание на всяка стъпка от алгоритъма, е необходимо да се зададе само как да се реагира на различните събития (действия на потребителя). В резултат програмистът създава не една голяма програма, а приложение, състоящо се от набор взаимосвързани процедури, управлявани от потребителя. В приложенията изпълнението на програмата не следва предварително определен път. Вместо това програмата обработва различни части код в отговор на настъпило събитие.

VisualBasic е един от първите езици, поддържащи събитийно-управляемо програмиране (event-driven programming). Този стил се съгласува отлично със стандарта GUI. Съществуват и други визуални езици за програмиране с подобен интерфейс (Borland Delphi, Borland C++ Builder и др.).

При събитийното програмиране програмата е в аморфно (изчакващо) състояние и настъпването на определени събития определя кога и коя част от програмния код ще бъде изпълнена.

В този вид програмиране изпълнението на програмата зависи от събитията, които ще възникнат.

Реакцията на определено събитие чрез изпълнение на програмен код не е задължителна. Реакцията на събитие, за което липсва програмен код, е стандартна, съгласно общоприетите разбирания на ОС.

При събитийното програмиране класовете от обекти имат допълнителна характеристика: набор от събития, на които техните екземпляри имат право да реагират.

II. Основни понятия

1. **Събитие** – действие, което протича в реално време и като отговор предизвиква едни или други такива. При събитийното програмиране програмистът има за задача да кодира определени действия, които се извършват като отговор на дадено събитие.

- събития свързани с клавиатурата
- събития свързани с мишката
- програмни събития

2. **Инициатор** (генератор) на едно събитие може да бъде:

- извършване на определено действие от страна на лицето, което използва (изпълнява) програмата – натискане на клавиш, придвижване или щракване на мишката и др.
- ОС – оповестяване, че текущият сеанс на работа с компютъра завършва и др.
- езиковия процесор – изтичане на зададен период от време.
- програмният код – изпълнение на специфични оператори или прилагане на определени методи към даден екземпляр на обект.

3. **Обект** - Във VisualBasic и другите среди за визуално програмиране обектите са елементи – бутони, рамки, скролери, полета и т.н. – т.е. това, което е създадено за използване в програмите. Обектите са елементите, от които се изгражда нашата програма. Обект може да бъде наречен както прост елемент (напр. Рамка), така и сложна форма, която съдържа няколко елемента. Обособена част от програмата също може да бъде наречена обект. Благодарение на средствата за визуална обработка можем да работим с обектите, като ги държим пред очите си и да получим резултата веднага. След като обектът е поставен във формата на средата за визуално програмиране, всичките му атрибути – размер, местоположение, цвят и др. –

се отразяват като програмен код, съответстващ на обекта. Програмистът има възможност да влияе пряко върху поведението на обектите, променяйки атрибутите им в хода на изпълнение на програмата, използвайки за целта написан програмен код.

4. **Свойство** (Атрибут) - Свойствата определят характеристиките на обекта и са негова неделима част. Свойствата се отнасят към обекта както локалните променливи се отнасят към процедурата. Изменението на свойствата влияе директно на обекта. Ние можем да влияем на обект, като променяме неговите свойства. Това може да стане както в стадия на писане на програмата, така и по време на нейното изпълнение.

5. **Обекти и събития. Обработчици на събития.** - Събитията възникват при действие от страна на потребителя или системата. Тази част от програмния код, която съответства на конкретно събитие, наричаме обработчик на събитие (манипулатор на събитие). Като правило всички обекти могат да реагират на събития – едни на повече, други на по-малко. Ние имаме възможност да изберем следните действия:

- Да игнорираме събитието – поведението на обекта се определя по подразбиране
- Да прихванем събитието – свързваме със събитието негов обработчик.

6. **Методи** - функционални характеристики, които определят “поведението” на обектите от всеки клас. Методите позволяват да се извършват определени операции над обектите.

7. Методи за работа с управляващи елементи (контроли).

А.Универсални

1. Drag действие – започва, завършва или прекъсва влачене

действие: 0-cancel (прекъсва); 1-започва; 2- завършва

2. Move (ляво, горно, широчина, височина) – премества управляващия елемент и/или променя размерите му

3. Refresh – принуждава да се извърши прерисуване на обекта

4. SetFocus –премества фокуса върху обекта, т.е. прави обекта активен

5. ShowWhatsThis – показва избраната точка от помощната информация
6. Zorder позиция – поставя обекта отпред или отзад

позиция: 0- отпред; 1- отзад;

Б. За ListBox (списъчна кутия) и ComboBox (комбинирана кутия)

1. AddItem – добавя точка от списък
2. RemoveItem – премахва точка от списък
3. Clear – изчиства съдържанието на управляващия елемент

II. Методи за работа с формуляри (форми)

А. За чертане

1. Circle – чертае окръжност, елипса или дъга
2. Line – чертае линия или правоъгълник
3. Pset – поставя точка с определен цвят
4. Point – връща като цяло число от тип Long RGB цвета на определена точка от формата или обекта PictureBox
5. PaintPicture - Рисува съдържанието на графичен файл.
6. Cls- Изчиства графика и текст, създадени по време на изпълнение

Б. За визуализация

1. Hide - скрива формата, но не я разтоварва (unload) от паметта
2. Show - показва формата

В. Други

PrintForm – изпраща за печат формата бит по бит

III. Методи за работа със системния буфер (Clipboard)

SetText Поставя копие на селектирания текст в буфера

GetText() Поставя копие на тестовото съдържание на буфера в указаната променлива

SetData Поставя копие на селектираната графика в буфера

GetData() Поставя копие на графиката от буфера в указаната променлива

GetFormat() Определя дали буферът съдържа текст или графика

Clear Изтрива съдържанието на буфера