

Introducere in Delphi

1. Introducere

Lucrarea este destinata celor care doresc sa dezvolte aplicatii Windows utilizand mediul de programare Delphi. Am ales Delphi-ul deoarece este usor de invatat, fiind foarte asemanator cu limbajul Pascal care este predat in liceu. Un alt motiv pentru care am ales mediul Delphi este ca in urma compilarii programelor se obtin executabile mici ce ruleaza rapid si nu necesita librarii ca sa poata rula (un program similar scris in Visual Basic necesita cca. 5 Megabytes de librarii).

Pentru a putea intelege si folosi acest manual este necesar ca utilizatorul sa cunoasca bine limbajul Pascal, deoarece Delphi-ul este practic limbajul Pascal orientat pe obiect. Cu toate ca Delphi-ul este un limbaj orientat pe obiect nu este necesara cunoasterea principiilor programarii intr-un astfel de limbaj pentru realizarea unor programe simple. Daca se doreste realizarea unor componente pentru Delphi sau modificarea unora dintre cele existente atunci este necesara cunoasterea principiilor programarii intr-un limbaj orientat pe obiect.

Am ales pentru prezentare Delphi 5 Enterprise Edition deoarece este varianta cea mai completa, celelalte variante :Professional Edition, Standard Edition fiind variante prescurtate, care nu contin toate facilitatile variantei Enterprise.

- CPU - Pentium 90 MHz, recomandabil 166 MHz
 - Windows 9x/Me, 2000, NT4 cu Service Pack 3 sau mai nou
 - Memorie - 32 Mb RAM, recomandabil 64 Mb
 - HDD - 80 Mb (instalare minima), recomandabil 250 Mb
 - Monitor VGA, recomandabil SVGA cu rezolutie de 800x600
 - Mouse
- Resursele necesare instalarii si rularii Delphi-ului sunt:
- CD-ROM

2. Prezentarea mediului DELPHI 5

La pornirea Delphi-ului acesta va arata ca in figura de mai jos:

Bara de meniuri Bare butoane Paleta de componente Pagini ale paletii de componente

Inspectorul de obiecte Fereastra proiectului nou inceput

Din bara de meniuri pot fi accesate meniurile Delphi-ului, bara de butoane permitand accesul rapid la principalele comenzi. Paleta de comenzi permite accesul rapid la cele peste 200 de componente ale mediului Delphi. Pentru o mai buna gestionare a componentelor acestea sunt impartite pe mai multe pagini, existand si posibilitatea adaugarii unor pagini noi si a unor componente noi. In partea stanga a imaginii poate fi observat inspectorul de obiecte (object inspector)

cu ajutorul caruia pot fi modificate proprietatile obiectelor (Properties) si pot fi stabilite ce rutine (proceduri) vor fi executate in momentul aparitiei unor evenimente (Events).

2.1 Principalele meniuri si comenzi (bara de meniuri)

Meniul File

Se utilizeaza a deschide, salva, inchide, tipari proiecte sau fisiere noi sau existente, pentru adaugarea unor noi ferestre(form) sau unitati de program(unit) proiectului deschis In meniu putem gasi urmatoarele optiuni:

- New - deschideofereasta de dialog de unde putem alege ce obiect dorim sa cream, sau putem selecta un vrajitor (wizard) care sa ne ajute la crearea unui nou obiect.
 - New Application - creeaza un nou proiect care contineofereasta goala, un unit si un fisier de proiect.
 - New Form - creeazaofereasta goala siadauga proiectului curent
 - New Frame - creeaza un nou frame (fereasta invizibila destinata componentelor nevizuale aflate in proiect, se utilizeaza pentruomai buna gestiune a componentelor folosite in cadrul proiectului) pentru proiectul curent
 - Open - deschideofereasta de dialog pentru incarcarea unui proiect, fereasta, unit sau fisier text.
 - Open Project - deschideofereasta de dialog pentru incarcarea unui proiect existent (fisiere cu extensia BPR sau BPK)
 - Reopen - deschide un meniu care contine cele mai noi proiecte sau module utilizate
 - Save - salveaza fisierul curent
 - Save As - salveaza fisierul curent utilizand un nume nou
 - Save Project As - salveaza proiectul curent cu un nume nou
 - Save All - salveaza toate fisierele deschise
 - Close - inchide proiectul curent si toate fisierele asociate
 - Close All - inchide toate fisierele deschise
 - Use Unit - adauga unit-ul selectat la clauza uses a modulului activ
 - Print - tipareste fisierul curent
 - Exit - inchide proiectul si paraseste mediul Delphi
- Meniul Edit

Se utilizeaza pentru a manipula text sau componente in timpul crearii programului. In meniu pot fi gasite urmatoarele optiuni:

- Undo/Undelete - anuleaza ultima actiune sau stergere
- Redo - anuleaza ultima operatie de undo
- Cut - muta obiectul selectat in Clipboard
- Copy - copiaza obiectul selectat in Clipboard
- Paste - copiaza continutul Clipboard-ului la pozitia cursorului
- Delete - sterge obiectul selectat
- Select All - selecteaza toate componentele de pe fereasta (form)
- Align to Grid - Aliniaza componentele selectate la cel mai apropiat punct din grila
- Bring to Front - muta componenta selectata in fata
- Send to Back - muta componenta selectata in spate
- Align - aliniaza componentele
- Size - redimensioneaza componentele selectate

•Scale - redimensioneaza proportional toate componentele din fereastra •Tab Order - modifica ordinea in care se sare apasand tasta Tab •Creation Order - modifica ordinea in care sunt create componentele nevizuale

•Flip Children - muta obiecte "in oglinda" e util pentru cazul in care nu se citeste de la stanga la dreapta ci invers

•Lock Controls Secures - "intepeneste" toate componentele in pozitiile curente

•Add to interface - defineste noua metoda, un nou eveniment, noua proprietate pentru componenta ActiveX

Meniul Search

Meniul se utilizeaza pentru a cauta text, obiecte, erori, unit-uri, variabile si simboluri in editorul de cod(fereastra in care se editeaza codul). In acest meniul putem gasi optiunile:

•Find - cauta un text specificat si marcheaza prima aparitie a acestuia in editorul de cod •Find in Files - cauta textul specificat si afiseaza fiecare potrivire in fereastra aflata sub

•Replace - cauta textul specificat si-l inlocuieste cu un altul •Search Again - repeta ultima cautare editorul de cod

•Incremental Search - cauta textul pe masura ce este scris •Go to Line Number - muta cursorul la linia specificata •Find Error -cauta ultima eroare de rulare

•Browse Symbol - cauta dupa un anumit simbol

Meniul View

Se utilizeaza pentru a afisa sau ascunde diverse elemente ale mediului Delphi. In meniul putem gasi urmatoarele optiuni:

•Project Manager - afiseaza managerul de proiect, acesta poate fi utilizat pentru a naviga prin fisierele proiectului, pentru a le copia, sterge, adauga, salva.

•Translation Manager - afiseaza managerul pentru traduceri

•Object Inspector - afiseaza inspectorul de obiecte •To-Do List - face vizibila lista cu ceea ce trebuie facut in proiect

•Alignment Palette - afiseaza fereastra in care se gasesc optiuni destinate alinierii obiectelor

•Browser - afiseaza fereastra in care pot fi vizualizate unit-urile si clasele utilizate in proiect

•Code Explorer - face vizibil editorul de cod

•Component List - afiseaza lista cu toate componentele Delphi-ului

•Window List - face vizibila lista cu toate ferestrele Delphi-ului care sunt deschise

•Debug Windows - afiseaza un meniul din care pot fi alese ce ferestre pentru depanarea programului vor fi afisate (puncte de intrerupere, apeluri ale subprogramelor, variabile, fire de executie, module, CPU, FPU)

•Desktops - permite afisarea, salvarea sau stergerea diverselor moduri de asezare a ferestrelor

•Toggle Form/Unit - comuta intre fereastra si unit-ul corespunzator •Units - afiseaza lista cu toate unit-urile proiectului •Forms - afiseaza lista cu toate ferestrele(form) existente in proiect

•Type Library - afiseaza editorul pentru librarii de tipuri (Type Library), util pentru a crea tipuri de informatii pentru controale ActiveX, obiecte MTS si alte obiecte COM

•New Edit Window - deschide inca un editor de cod

•Toolbars - deschide un meniul din care pot fi selectate ce bare de butoane vor fi afisate

Meniul Project

Este utilizat pentru a compila o aplicatie. Pentru ca optiunile din meniu sa poata fi folosite e necesar sa fie deschis un proiect.

In meniu avem urmatoarele comenzi:

- Add to Project - adauga un fisier proiectului
- Remove from Project - elimina un fisier din proiect
- Import Type Library - afiseaza o lista cu librariile de tipuri inregistrate in sistem
- Add to Repository - adauga proiectul la lista de obiecte ce pot fi create, din meniul File optiunea New, pentru a fi adaugate la proiect.
- View Source - afiseaza sursele proiectului in editorul de cod
- Languages - permite adaugarea, eliminarea sau reinnoirea DLL-urilor pentru limba
- Add New Project - adauga la proiect o noua aplicatie, un nou DLL ...
- Add New Project - adauga la proiect un altul existent
- Compile project - compileaza doar acele fisiere care au fost modificate de la ultima
- Build project - compileaza toate modulele proiectului
- Syntax Check project - verifica sintactic toate modulele proiectului compilare
- Information for project - afiseaza informatii despre proiectul compilat (nr. de linii, dimensiunea programului rezultat ...)
- Compile All Projects - compileaza toate fisierele care au fost modificate de la ultima compilare in toate proiectele din grupul de proiecte
- Build All Projects - compileaza toate fisierele in toate proiectele din grupul de proiecte
- Web Deployment Option - se utilizeaza pentru a configura controalele ActiveX sau ActiveForm in vederea adaugarii lor la serverul web.
- Web deploy - adauga controalele ActiveX sau ActiveForm, dupa configurarea acestora
- Options - afiseaza o fereastra de dialog in care se modifica optiunile pentru compilare, link editare etc.

Meniul Run

Meniul permite depanarea programelor. In acest meniu se afla urmatoarele optiuni:

- Run - compileaza si executa aplicatia
- Attach to Process - afiseaza o lista cu procesele care pot fi depanate
- Parameters - de aici se pot specifica: parametrii cu care va fi lansata in executie aplicatia, executabilul corespunzator unui DLL, un calculator pentru depanarea la distanta
- Register ActiveX Server - adauga o intrare in registrul Windows-ului pentru controlul ActiveX, e valabila comanda atunci cand avem un proiect ActiveX.
- Unregister ActiveX Server - sterge intrarea in registrul Windows-ului pentru controlul ActiveX, e valabila comanda atunci cand avem un proiect ActiveX.
- Install MTS Object - Instaleaza obiectele MTS din proiect intr-un pachet MTS, comanda valabila doar daca proiectul curent este un obiect MTS (Microsoft transaction server)
- Step over - executa un program linie cu linie, sarind peste proceduri, tratandu-le ca fiind o singura instructiune
- Trace Into - executa un program linie cu linie, intrand in proceduri

- Trace to Next Source - executa un program, oprindu-se la urmatoarea linie executabila din program
- Run To Cursor - Executa programul curent pana la pozitia cursorului din editorul de cod

- Run Until Return - executa procesul pana la iesirea din functia curenta
- Show Execution Point - pozitioneaza cursorul la linia care se executa
- Program Pause - opreste temporar executia programului ce ruleaza
- Program Reset - termina executia programului ce ruleaza si elibereaza memoria ocupata de catre acesta
- Inspect - deschideofereastra in care putem scrie numele variabilei ce se inspecteaza
- Evaluate/Modify - afiseazaofereastra in care putem evalua sau modifica valoarea
- Add Watch - afiseazaofereastra in care putem scrie mai multe variabile a caror valori dorim sa le urmarimunei expresii.
- Add Breakpoint - afiseaza un meniu din care putem adauga, modifica puncte de intrerupere

Meniul Component

Din acest meniu se pot gestiona componentele instalate. Optiunile acestui meniu sunt:

- New Component - deschideofereastra cu ajutorul careia se pot crea noi componente
- Install Component - instaleazaonoua componenta existenta
- Import ActiveX Control - adaugaolibrarie de tipuri de controale ActiveX
- Create Component Template - salveaza componente modificate ca sabloane cu un nou
- Install Packages - instaleaza un pachet de componente
- Configure Palette - deschideofereastra din care se poate configura paleta de nume, onoua icoana, peonoua pagina componente, se pot sterge, ascunde, muta componente dintr-opagina in alta

Meniul Database

Din acest meniu se pot crea, modifica, urmari si vizualiza bazele de date. Meniul contine urmatoarele optiuni:

- Explore - lanseaza in executie Database Explorer sau SQL Explorer, cu ajutorul carora se pot crea si manipula baze de date
- SQL Monitor - lanseaza SQL Monitor cu ajutorul caruia se pot urmari alocarea resurselor si starea comenzilor date prin SQL Links
- Form Wizard - se utilizeaza pentru crearea rapida a unor ferestre de vizualizare a bazelor de date

Meniul Tools

Meniul tools se utilizeaza pentru:

- A vizualiza sau modifica configuratia mediului de programare
- A vizualiza sau modifica configuratia depanatorului
- A modifica obiectele ce pot fi create cu Delphi-ul
- A modifica lista programelor ce apar in acest meniu
- A crea sau modifica tabele ale bazelor de date
- A crea sau modifica pachete de componente
- A crea sau modifica imagini

In acest meniu putem avea:

- Enviroment Options - specifica configuratia dorita, caile catre librarii, paleta de
- Editor Options - de aici se configureaza editorul de cod (culori, taste rapide, indentari etc)componente
- Debugger Option - configurare program de depanare
- Repository - gestioneaza modul de organizare a obiectelor ce pot fi create cu optiunea
- Translation Repository - afiseaza translation repository, cu care por fi facute traduceri
- Configure Tools - cu ajutorul aceste optiuni pot fi adaugate, sterse sau modificate

New din meniul File

optiuni ce apar in acest meniu Unelte optionale:

- Database Desktop -lanseaza un utilitar cu ajutorul caruia putem crea, vizualiza, sorta, modifica si interoga baze de date.
- Package Collection Editor - editor de pachete

- Image Editor - lanseaza un utilitar cu ajutorul caruia putem edita fisiere de resurse,
- Team Source - lanseaza un program destinat gestionarii unor proiecte dezvoltate de catre grupuri de programatori.icoane, bitmap-uri si cursoare care sa fie folosite in aplicatii

Meniul Help

Se utilizeaza pentru a accesa help-ul sistemului. In acest meniu putem gasi urmatoarele optiuni:

- Delphi Help - deschide help-ul Delphi-ului
- Delphi Tools - deschide help-ul destinat uneltelor din Delphi.
- Windows API/SDK Help - deschide help-ul pentru apelurile API Windows
- Borland Home Page - salt la pagina web a Borland-ului (Inprise-ului)
- Delphi Home Page - salt la pagina web a Delphi-ului
- Delphi Developer Support - salt la pagina web destinata programatorilor
- Delphi Direct - legatura directa catre pagina Delphi-ului
- Customize - permite gestionarea help-ului
- About - informatii despre copyright si versiune

2.2 Bara de butoane

Pe bara de butoane se afla scurtaturi catre cele mai utilizate comenzi.

Open project Remove from project

New Open Reopen Save Save all Add to projectHelp

Units Forms Toggle New form Run Pause Trace into Step over

2.3 Paleta de componente

Paleta de componente a Delphi-ului este impartita in mai multe pagini, in interiorul fiecarei pagini existand un numar variabil de componente. Structura paginilor poate fi schimbata, utilizatorul avand posibilitatea sa creeze pagini noi, componente noi, poate sa stearga componente sau poate sa le reordoneze.

Pentru a obtine informatii generale despre pagina activa utilizatorul poate apasa tasta F1. Daca o componenta este selectata in momentul apasarii tastei F1 atunci vor fi afisate informatii despre componenta respectiva.

Componentele pot sa fie:

- Vizibile - componenta este vizibila, sau poate fi facuta vizibila in timpul rularii programului, spre exemplu butoanele

- Invizibile - componenta apare sub forma unei imagini la scrierea programului, dar in timpul rularii nu e vizibila, spre exemplu timer-ul

Fiecare componenta are anumite atribute cu ajutorul carora controlam aplicatia :

- Proprietatile(properties) - ne permit accesul la metodele sau campurile unei clase.

- Metode(methods) - proceduri si functii asociate unui obiect

- Evenimentele(events) - sunt actiuni ale utilizatorului, ex. apasarea unui buton,

sau ale sistemului de operare recunoscute de catre componenta. Fiecare componenta are lista de evenimente pe care le recunoaste si la care poate raspunde. Exista posibilitatea de a asocia un cod care sa se execute fiecarui eveniment. In tabelul de mai jos sunt prezentate paginile paletei de componente, precum si descriere scurta a continutului fiecarei pagini.

Nume

Descriere

Standard

Controale standard: butoane, meniuri, etichete

Additional

Controale specializate: butoane ce pot contine imagini, liste derulante

Win32

Controale pentru Windows9x/NT: calendar, bare de stare,

System

Componente si controale destinate lucrului cu sistemul: timer, media player

Data Access

Componente invizibile cu ajutorul carora se poate realiza conectarea la diferite baze de date utilizand Borland Database Engine (BDE)

Data Controls

Componente vizuale destinate accesului (vizualizare, stergere, modificare, adaugare date) la bazele de date

ADO

Componente ce permit conectarea la baze de date folosind ActiveX Data Objects (ADO)

Interbase

Componente ce permit conectarea la un server Interbase fara BDE sau ADO

Midas

Componentele din aceasta pagina permit lucrul cu baze de date distribuite

InternetExpress Componentele din aceasta pagina aplicatii InternetExpress care sunt simultan aplicatii Web Server si clienti pentru baze de date distribuite permit construirea de

Internet

Suport pentru crearea aplicatiilor pentru Internet

FastNet

Ofera o varietate de protocoale de acces la Internet

Decision Cube

Componente care vizualizeaza datele din bazele de date ajutand la luarea deciziilor

Quick Report

Componente ce ajuta la crearea unor rapoarte

Win3.1

Componente pentru compatibilitatea inapoi cu Delphi 1 (a nu se folosi)

Samples

Exemple de componente (sursele lor se gasesc: Delphi5sourcesamples)

ActiveX

Componente model de tip ActiveX

Servers

Componente destinate accesului la diferite servere

2.4 Inspectorul de obiecte (object inspector)

Inspectorul de obiecte este utilizat pentru a modifica proprietatile si a scrie rutinele de tratare a evenimentelor. Acesta este prevazut cu doua pagini: obiectelor

- Properties - care permite stabilirea
- Events - destinata gestionarii evenimentelor legate de componente, proprietatilor obiectelor

Pagina Properties

Este prevazuta cu doua coloane:

pe coloana stanga sunt listate denumirile proprietatilor

pe coloana dreapta avem valorile actuale ale acestora. Atribuirea unei valori se poate face in trei moduri:

prin linia de editare aferenta proprietatii(ex. Caption)

prin lista derulanta(ex. BorderStyle)

prin intermediul unei ferestre de dialog(ex. Font)

Anumite proprietati (ex. Color) admit doua modalitati de atribuire a unei valori, prin lista derulanta, sau daca se da dublu clic, prin fereastra de dialog.

La unele proprietati (ex. BorderIcons, Constrains) numele este precedat de un semn plus. In aceste cazuri proprietatea respectiva desemneaza o lista de proprietati.

Pagina Events

Este prevazuta cu doua coloane:

pe coloana stanga sunt listate denumirile evenimentelor

pe coloana dreapta avem procedurile ce trateaza evenimentele. Atribuirea unei valori se poate face in doua moduri:

prin linia de editare, se da dublu clic pe coloana din dreapta, corespunzatoare evenimentului pe care dorim sa-l tratam, iar Delphi-ul deschide editorul de cod si ne permite scrierea codului ce se doreste a fi executat la aparitia evenimentului respectiv. Antetul procedurii(metodei) de tratare a evenimentului este scris automat de catre editor. Daca procedura aferenta unui eveniment nu contine nici o linie de cod aceasta va fi stearsa.

prin lista derulanta, din care se poate alege o procedura care a fost definita de programator anterior, daca se doreste sa se execute aceeasi rutina pentru mai multe evenimente.

3. Gestiunea componentelor

3.1 Asezarea componentelor pe o forma(form)

Pentru a aseza o componenta se poate proceda astfel: - Se efectueaza un clic de mouse pe simbolul grafic al componentei

- Se efectueaza un clic in pozitia in care se doreste plasarea coltului stanga sus a componentei.

O alta posibilitate de a aseza o componenta, dar de aceasta data in centrul ferestrei este de a da dublu clic pe aceasta.

Daca se doreste adaugarea mai multor componente identice atunci se procedeaza astfel: -

Se apasa tasta SHIFT

- Se efectueaza clic pe simbolul grafic al componentei dorite, dupa ce se poate elibera tasta SHIFT

- Se pozitioneaza obiectul in pozitiile dorite

- Pentru revenirea la normal se va efectua un clic pe simbolul sagetii, aflat in partea din stanga a paginii cu componente.

Delphi-ul contine si anumite componente destinate gruparii mai multor componente: GroupBox, RadioGroup si Panel. Aceste componente contin permit

gestionare mai usoara a componentelor continute: se pot muta simultan, pot fi modificate anumite proprietati simultan(font, culoare)

3.2 Manipularea componentelor asezate

Operatiile se pot executa asupra unei componente doar daca aceasta este selectata. Putem selecta o componenta in mai multe moduri:

Se efectueaza clic in interiorul componentei

Se alege obiectul din lista derulanta a inspectorului de obiecte

Daca focarul este pe fereastra se poate trece de la componenta la alta cu tasta TAB In anumite cazuri este necesara realizarea unei operatii asupra mai multor componente.

Pentru aceasta este necesara selectarea obiectelor asupra carora se va efectua operatia. Putem selecta un grup de obiecte in doua moduri:

- Cu tasta SHIFT apasata executam clic pe fiecare din componentele dorite
- Realizam un dreptunghi imaginar prin apasarea butonului stang al mouse-ului si miscarea acestuia astfel incat sa cuprindem in interiorul dreptunghiului componentele dorite.

Daca de doreste selectarea tuturor obiectelor se poate alege optiunea Select All din meniul Edit.

- Cut - muta obiectele selectate in clipboard
 - Copy - copiaza obiectele selectate in clipboard
 - Paste - copiaza din clipboard
- Asupra obiectelor selectate se pot aplica comenzile(ce se gasesc in meniul Edit):
- Delete - sterge obiectele selectate
 - Align to Grid - aliniaza toate componentele la cele mai apropiate puncte de retea
 - Bring to Front - deplaseaza componentele selectate in fata celorlalte componente
 - Send to back - deplaseaza componentele selectate in spatele celorlalte componente
 - Align - afiseaza o fereastra de dialog cu ajutorul careia un grup de componente pot fi
 - Size - afiseaza o fereastra de dialog cu ajutorul careia se pot redimensiona componentele selectate(la dimensiunea celei mai mari, celei mai mici componente ...)aliniate (dupa cea mai din stanga, cea mai din dreapta, cea mai de sus ...)

- Scale - se pot scala (mari sau mica) procentual componentele selectate

•Tab Order - afiseaza o fereastra de dialog cu ajutorul careia putem schimba ordinea de parcurgere a componentelor, acesta se parcurg in ordine cu Tab sau cu Shift+Tab in ordine inversa

•Creation Order - afiseaza o fereastra de dialog din care putem schimba ordinea de creare a obiectelor

•Lock - blocheaza/deblocheaza o componenta

Comenzile Align to Grid, Bring to Front, Send to Back, Align, Size, Scale, Tab Order, Creation Order mai pot fi gasite si in meniul care apare daca executam clic dreapta pe un obiect.

Daca se doreste se poate face vizibila o paleta de comenzi destinate alinierii obiectelor(meniul View optiunea Alingment Palette).

4 Programarea in Delphi

4.1 Lectia 1 - Programarea orientata pe obiect (OOP)

Notiunea de obiect. Caracteristici.

Ideea OOP este de a pune la un loc, intr-un singur container, atat datele cat si partea de cod(proceduri si functii, care poarta denumirea de metode) . Containerul se numeste obiect.

Obiectul releva utilizatorului (programatorului care utilizeaza obiectul) doar ceea ce este necesar, astfel incat utilizarea obiectului sa fie cat mai usoara. Toate mecanismele interne pe care utilizatorul nu are nevoie sa le cunoasca sunt ascunse. OOP permite abordarea modulara a programarii, ceea ce ne fereste de necesitatea rescrierii unor parti de cod.

O alta caracteristica importanta a obiectelor este mostenirea. Prin mostenire se intelege acea proprietate a obiectelor prin care un obiect nou construit preia datele si metodele unui alt obiect, parinte. Aceasta caracteristica ne permite scrierea rapida si usoara a unor obiecte asemanatoare, care au unele date si metode comune.

Se poate intampla ca anumite metode a unei clase sa fie rescrise, astfel incat, chiar daca oclasa descendenta are metode cu acelasi nume, sa faca cu totul altceva. Aceasta caracteristica poarta denumirea de polimorfism.

Notiunea de clasa. Caracteristici.

O clasa sau un tip de clasa definesc structura ce contine campuri, metode si proprietati. Instanta unui tip de clasa poarta denumirea de obiect.

Un camp este de fapt variabila care face parte dintr-un obiect. La fel ca si campurile unui record, campul unei clase reprezinta odata care exista in fiecare instanta a unei clase. O metoda este o procedura sau o functie asociata cu o clasa. Majoritatea metodelor opereaza asupra obiectelor, care sunt instante ale claselor.

O proprietate este o interfata pentru date asociata unui obiect (date care de obicei este stocate intr-un camp). Proprietatile au specificatori de acces care determina modul in care datele pot fi citite sau modificate. Din program, din afara obiectului, o proprietate apare cel mai adesea la fel ca un camp.

Pentru obiecte se alocă dinamic blocuri de memorie a caror structura este determinată de tipul de clasa. Fiecare obiect are o copie unică a fiecărui camp definit in clasa, dar toate instantele unei clase impart aceleasi metode. Obiectele sunt create si distruse cu ajutorul unor metode speciale numite constructori si destructori.

Constructorul este o metoda utilizata pentru crearea unei instante a unei clase. Are rolul de a initializa unele campuri, de a alocă memorie, sau a face tot ceea ce este necesar initializării obiectului. Nu este obligatoriu sa definim un constructor, daca acesta nu a fost definit se utilizeaza constructorul implicit Create.

Destructorul este o metoda speciala care este apelata automat la distrugerea obiectului.

De obicei destructorul este vazut ca fiind inversul constructorului, el ocupandu-se de dezallocarea memoriei alocate clasei. Nu este obligatoriu ca fiecare clasa sa aiba un destructor.

Vizibilitatea membrilor unei clase.

Fiecare membru a unei clase are un atribut denumit vizibilitate, care este indicata de unul dintre cuvintele rezervate: private, protected, public, published. Vizibilitatea determina modul in care membrii pot fi accesati:

In Pascal standard notiunea de clasa nu exista, ea fiind inlocuita cu cea de obiect, de aici pot apărea anumite confuzii. Definitia de mai sus e valabila pentru Delphi, nu si pentru Pascal.

- Private - un membru private este invizibil in afara clasei in care este declarat, altfel spus o metoda declarata ca fiind private nu poate fi apelata din alt modul, iar un camp sau proprietate nu pot fi citite sau scrise din alt modul

- Protected - un membru protected este vizibil doar in clasa in care a fost declarat si in toate clasele descendente.

- Public - un membru public este vizibil in exteriorul clasei.

- Published - acest mod de acces este utilizat atunci cand se scriu componente. Toti membrii declarati in aceasta zona vor fi vizibili in Object Inspector.

In scrierea programelor Delphi este utilizata foarte mult OOP deoarece se utilizeaza in

special componente. Aceste componente sunt componente!) care fie sunt create de utilizator, fie sunt venite impreuna cu Delphi-ul, si pot fi reutilizate in orice alt program. obiecte(dar nu toate obiectele

Prezint mai jos componenta TAnimate2 derivata din TAnimate :

```
type
TAnimate2 = class(TAnimate) private
{ Private declarations, tot ceea ce e declarat mai jos nu e vizibil in afara clasei}
fhintimage:Tbitmap;
fhintcolor:Tcolor; fshowimage:boolean; { Proceduri si functii }
procedure SetHintImage(v:Tbitmap); procedure SetHintColor(v:Tcolor);
procedure setshowimage(v:boolean); procedure afisimg(Sender:TObject);
procedure CMHintShow(var Message: TMessage); message CM_HINTSHOW; protected
{ Protected declarations } public
{ Public declarations }
constructor Create(AOwner:TComponent);override; destructor Destroy;override;
published
{ Published declarations }
{pentru a citi proprietatea HintImage se citeste din variabila fhintimage, iar pentru
aoscree se foloseste procedura SetHintImage}
property HintImage:Tbitmap read fhintimage write SetHintImage;
{proprietatea de mai jos are definita siovaloare implicita(clwhite - culoarea alba)} property
HintTransparentColor:Tcolor read fhintcolor write SetHintColor default clwhite; property
ShowHintImage:boolean read fshowimage write SetShowImage default false; property active;
property align; property Anchors; property Autosize;
property BorderWidth; property Center;
property Color; property CommonAVI; property Constraints;
property Cursor; property FileName; property Height;
property HelpContext; property Hint;
property Left;
```

```
property Name; property ParentColor;
property ParentShowHint; property Repetitions;
property ShowHint; property StartFrame;
property StopFrame; property Tag;
property Timers; property Top;
property Transparent; property Visible;
property Width; {*****Evenimente} property OnClose;
property OnOpen; property OnStart;
property OnStop; property OnClick;
property OnMouseMove; end;
```

In exemplul de mai sus ati observat ca a aparut override, care ne indica ca acea metoda poate fi rescrisa, astfel incat intr-oclasa derivata metoda respectiva va putea executa un cu totul alt cod. Daca se doreste ca utilizatorul sa fie obligat sa rescrie metoda atunci se poate adauga virtual in dreptul metodei.

Intrebari si probleme

1. Ce este un obiect?
2. Cum pot ascunde metoda, dar clasele derivate sa poata apela?
3. Ce este clasa?
4. Care este diferenta dintre clasa si obiect?
5. Din ce cauza avem metode si campuri private?

6. Cand este apelat destructorul unei clase?

4.2 Lectia 2 - Form(TForm)

Pentru descrierea formelor pot fi utilizate 50 de proprietati, care pot fi stabilite in perioada de elaborare a proiectului prin intermediul inspectorului de obiecte. Formele pot sa "reactioneze" la 34 de evenimente. Scopul acestui capitol este prezentarea celor mai importante proprietati, metode si evenimente referitoare la forme. Se mentioneaza faptul, ca majoritatea proprietatilor si evenimentelor formelor sunt identice cu proprietatile si evenimentele referitoare la majoritatea componentelor. Astfel, de exemplu, atributul Width al unei forme descrie latimea formei, iar in context cu componenta acest atribut se va referi la latimea componentei. Evenimentul OnClick in contextul formelor se refera la efectuarea unui clic pe forma, iar in context cu componentele evenimentul se refera la efectuarea unui clic pe componenta.

Proprietatile formelor

Proprietatea Action

Este utilizata pentru centralizarea raspunsurilor la comenzile date de utilizator. Majoritatii componentelor li se poate asocia o actiune, actiunile fiind centralizate si gestionate folosind componenta ActionListener din pagina standard.

Proprietatea ActiveControl

Este utilizata atunci, cand pe forma au fost asezate mai multe componente si se doreste specificarea componentei care sa fie in focar, in momentul initial. Prin efectuarea unui clic pe butonul "sageata jos" situat in coloana de valori se poate deschide o lista, care contine numele tuturor componentelor formei, care pot fi in focar. Din lista afisata, printr-un clic, se poate selecta componenta necesara. La un moment dat numai o singura componenta poate sa fie in focar. Proprietatea ActiveControl poate fi utilizata si in executie. Daca in aceasta proprietate se incarca un alt nume de componenta, atunci componenta respectiva va fi pusa in focar. Prin testarea valorii atasate acestei proprietati se poate decide care anume componenta este in focar la un moment dat. Daca forma nu este in focar, componenta memorata in ActiveControl desemneaza acea componenta, care va ajunge in focar in momentul in care forma va reobține focarul.

Proprietatea Align

Se utilizeaza pentru a alinia componentele sus, la stanga, jos sau la dreapta pe forma sau panel, acestea ramanand pe pozitie chiar daca forma sau panelul se redimensioneaza. Valoarea implicita este alNone, ceea ce inseamna ca obiectul ramane acolo unde a fost pozitionat.

Proprietatea Anchors

Se utilizeaza pentru a ne asigura ca obiectul isi pastreaza pozitia relativ la marginile parintelui (a componentei care o contine), chiar daca componenta parinte este redimensionata. Cand parintele este redimensionat obiectul isi pastreaza pozitia relativ la marginile de care a fost ancorat.

Proprietatea AutoScroll (boolean)

Se refera la modul de utilizare a barelor de defilare atasate formei. In cazul in care se alege valoarea True, care este si valoarea implicita, atunci in caz de nevoie, in executie, forma va fi prevazuta cu bare de defilare. Aceste bare vor apare atunci cand utilizatorul a redimensionat forma prin care componentele asezate pe forma devin invizibile. Daca

se alege valoarea False, nu se vor aseza in mod automat bare de defilare la forma redimensionata. In acest caz atasarea barelor de defilare poate fi realizata prin intermediul proprietatilor HorzScrollBar.

Proprietatea `AutoSize`(boolean)

Se utilizeaza atunci cand se doreste ca obiectul sa se redimensioneze automat. Daca e pusa pe `true` atunci se redimensioneaza automat daca continutul se schimba.

Proprietatea `BidiMode`

Permite ajustarea modului de afisare si a comportamentului automat atunci cand aplicatia ruleaza intr-ozona in care se citeste de la dreapta la stanga si nu de la stanga la dreapta.

Proprietate compusa `BorderIcons`

Se refera la existenta in linia de titlu a formei meniului sistem si a butoanelor de minimizare si maximizare. Aceasta proprietate este de tip multime, valorile posibile fiind urmatoarele:

Valoare

Semnificatie

`biSystemMenu`

Forma este prevazuta cu meniul sistem (control)

`biMinimize`

Forma este prevazuta cu un buton de minimizare

`biMaximize`

Forma este prevazuta cu un buton de maximizare

`biHelp`

Daca proprietatea `BorderStyle` contine valoarea `bsDialog` sau daca sunt excluse valorile `biMinimize` si `biMaximize`, atunci in bara de titlu va apare un semn de intrebare. Daca se efectueaza un clic pe acest semn, cursorul de mouse va avea forma definita de comanda `crHelp`.

Valoarea implicita a proprietatii `BorderIcons` este (`biSystemMenu`, `biMinimize`, `biMaximize`).

Proprietatea `BorderStyle`

Se refera la stilul chenarului formei. Valorile posibile ale acestei proprietati sunt urmatoarele:

Valoare

Semnificatie

`bsDialog`

Chenar neredimensionabil; chenarul standard al ferestrelor de dialog.

`bsSingle`

Chenar neredimensionabil, desenat cu linie simpla.

`bsNone`

Chenar neredimensionabil, desenat cu linie invizibila.

`bsSizeable`

Chenar redimensionabil standard.

`bsToolWindow`

Similar cu `bsSingle`, dar este prevazut cu un mic titlu.

`bsSizeToolWin`

Similar cu `bsSizeable`, dar este prevazut cu un mic titlu.

Proprietatea `BorderWidth`(de tip word)

Se utilizeaza pentru controlul bordurii. Bordura este zona din imediata apropiere a marginilor in care niciocomponenta nu poate fi plasata.

Proprietatea `Capiton` (de tip string)

Specifica un sir de caractere care este afisat in linia de titlu al formei, de exemplu Form 1. In cazul componentelor aceasta proprietate desemneaza eticheta atasata componentei. In acest ultim caz litera precedata de caracterul "&" este litera de selectie a componentei (litera de selectie este subliniata si componenta poate fi selectata prin combinatia Alt - litera subliniata).

Proprietatile ClientHeight si ClientWidth (de tip integer)

Se refera la inaltimea si latimea zonei client a formei. Aceste valori sunt specificate in puncte. Zona client este aria utilizabila in interiorul chenarului formei.

Proprietatile Color (de tip Tcolor=longint)

Stabileste culoarea de fond a formei. Pentru initializarea valorii acestei proprietati pot fi folosite constante predefinite sau valori returnate de functia RGB. Valoarea implicita este clBtnFace. Daca pentru proprietatea ParentColor corespunzatoare unei componente se ia valoarea True, si se modifica valoarea proprietatii Color corespunzatoare formei proprietar al componentei, atunci se va modifica automat si culoarea componentei.

Proprietatea Constraints

Se utilizeaza pentru a stabili dimensiunea minima si maxima pentru componenta. Aceste dimensiuni nu pot fi depasite.

Proprietatea Ctl3D (de tip boolean)

Determina daca forma sau componenta are aspect tridimensional (pentru True) sau aspect bidimensional (pentru False). Valoarea implicita este True. Daca pentru proprietatea ParentCtl3D corespunzatoare unei componente se ia valoarea True, si se modifica valoarea proprietatii Ctl3D corespunzatoare formei proprietar al

componentei, atunci se va modifica automat si aspectul componentei.

Proprietatea Cursor

Stabileste imaginea utilizata pentru reprezentarea cursorului mouse-ului.

Pentru selectarea unei imagini se vor folosi diferitele constante predefinite de descriere a cursorului. Valoarea implicita este constanta crDefault.

Proprietatea DefaultMonitor

Stabileste pe care monitor va fi afisata forma, in cazul aplicatiilor multimonitor.

Proprietatea DockSite

Stabileste daca forma poate fi tinta unei operatii drag-and-dock(trage si stabileste)

Proprietatea DragKind

Se foloseste pentru a stabili daca se iau obiecte de pe control(drag) sau se pot pune pe el(dock).

Proprietatea Enabled (de tip boolean)

Se refera la accesibilitatea formei sau componentei. Daca se alege varianta True, care este si valoarea implicita, atunci forma sau componenta va reactiona la diferitele evenimente referitoare la mouse, tastatura si ceas. Daca se alege varianta False, forma sau componenta va fi inhibata si nu va raspunde la evenimentele echipamentelor de intrare. Valoarea acestei proprietati de regula este modificata in executie, de exemplu pentru a interzice utilizarea unor butoane de comanda in contextul respectiv. Eticheta unei componente inhibate este afisata cu caractere estompate. O componenta inhibata ulterior poate fi reactivata, moment in care eticheta componentei va fi afisata cu caractere negre.

Proprietatea compusa Font

Stabileste fontul utilizat la textele care sunt afisate pe obiectul Canvas al formei precum si la textele aferente componentelor asezate pe forma. Fontul utilizat poate fi definit in perioada de elaborare a proiectului, insa poate fi modificat in executie, prin redefinirea subproprietatilor Color (culoare), Height (inaltime), Name (nume), Pitch (latimea caracterelor), Size (dimensiune), sau Style (stil).

Proprietatea FontStyle

Stabileste stilul formei. Valorile posibile ale acestei proprietati sunt urmatoarele:

Valoare

Semnificatie

fsNormal

Forma creaza o fereastră normală (nici MDI copil, nici MDI parinte). Valoare implicita.

fsMDIChild

Forma creaza o fereastră MDI copil

fsMDIForm

Forma creaza o fereastră MDI parinte

fsStayOnTop

Forma va fi afisată deasupra celorlalte forme utilizate în proiect

Proprietatile Height si Width (de tip integer)

Sabesc inaltimea, respectiv latimea formei sau componentei. Aceste valori sunt exprimate in puncte care include si dimensiunile rezervate chenarelor.

Proprietatea HelpContext

Se foloseste pentru a stabili un numar folosit in apelarea ajutorului (Help) context-senzitiv, pentru a determina care fereastră de ajutor este afisata daca utilizatorul apasa tasta F1. Daca HelpContext are valoarea 0 atunci va fi afisata fereastră de ajutor corespunzatoare parintelui.

Proprietatea HelpFile

Specifica fisierul in care se afla ajutorul.

Proprietatea Hint (de tip string)

Specifica un text explicativ care va fi afisat atunci cand cursorul de mouse este asezat deasupra unei forme sau componente. Acest text este afisat numai atunci cand proprietatea ShowHint (de tip boolean) este initializata cu valoarea True.

Proprietatile compuse HorzScrollBar si VertScrollBar

Stabesc daca forma este prevazuta sau nu cu bare de defilare orizontale si/sau verticale. Subproprietatile barelor de defilare sunt centralizate in tabelul urmator:

Proprietate	Descriere
-------------	-----------

Increment	
-----------	--

(o valoare cuprinsa intre 1 si 32767, valoarea implicita este 8)	Stabileste distanta de deplasare a imaginii din forma in momentul in care utilizatorul efectueaza un clic pe unul din butoanele cu sageti situate in extremitatile barelor de defilare.
--	---

Margin (de tip implicita este 0) word, valoarea implicita este 8)	Stabileste momentul de afisare al barei de defilare. Aceasta proprietate determina numarul minim de puncte care separa componentele formei si muchia formei. In executie, daca o componenta ajunge la o distanta mai mica decat Margin
---	--

puncte de la muchie, si daca proprietatea Visible are valoarea True, atunci se afiseaza bara de defilare.	
---	--

Position	
----------	--

(de tip integer, valoarea implicita este 0) Comunica pozitia actuala a cursorului glisant al barei de defilare. Aceasta valoare este actualizata automat in executie, in functie de defilare.

Range

(de tip integer, valoarea implicita este 0) Dimensiunea maxima de defilare, exprimata in puncte. La bara de defilare verticala aceasta valoare trebuie sa fie mai mare decat valoarea proprietatii Height (in caz contrar bara verticala este invizibila). La bara

de defilare orizontala aceasta valoare trebuie sa fie mai mare decat valoarea proprietatii ClientWidth (in caz contrar bara orizontala este invizibila). De exemplu, daca proprietatea Range a unei bare orizontale contine valoarea 500, latimea formei este 200, atunci valorile proprietatii Position pot fi in intervalul (0, 300).

Tracking (de avand valoarea implicita False) tip boolean, Daca se alege valoarea True, imaginea din forma se deplaseaza impreuna

cu butonul de glisare tras de utilizator. Daca se alege valoarea False, imaginea ramane statica atata timp cat utilizatorul trage butonul de glisare, deci actualizarea imaginii defilate va avea loc numai in momentul eliberarii butonului glisant al barei de defilare.

Visible (deav nd valoarea implicita True) tip boolean, Se refera la vizibilitatea barei de defilare. Daca se alege valoarea True, si
daca componenta se afla la o distanta mai mica decat Margin puncte de muchia formei, atunci bara de defilare devine vizibila.

Proprietatea Icon

Specifica simbolul grafic care va fi utilizat atunci cand forma este minimizata.

Proprietatea KeyPreview (de tip boolean)

Se refera la modul de gestiune a evenimentelor referitoare la utilizarea tastaturii (OnKeyDown, OnKeyPress si OnKeyUp). Daca pentru aceasta proprietate se alege valoarea True, atunci evenimentele referitoare la tastatura sunt gestionate de forma actuala, si nu de componentele asezate pe forma, chiar daca in momentul respectiv in focar se afla o componenta. De exemplu, sa presupunem ca forma contine un buton de comanda care este in focar. Daca utilizatorul apasa o tasta, evenimentele aferente tastarii vor fi gestionate de procedurile formei si nu de procedurile butonului de comanda. Daca proprietatea este initializata cu valoarea False, care este si valoarea implicita, atunci evenimentele referitoare la tastare vor fi gestionate de procedurile componentei active, si nu de procedurile formei.

Proprietatile Left si Top (de tip integer) Stabilesc abscisa, respectiv ordonata:

- Formei fata de coltul stanga sus al ecranului;
- Componentei fata de coltul stanga sau al formei proprietar;
- Componentei fata de coltul stanga sus al unei componente de tip container. Valorile acestor proprietati sunt date in puncte.

Proprietatea Menu (de tip TmainMenu)

Stabileste componenta de meniu principal care este utilizat de forma la un moment dat. In perioada de elaborare a proiectului aceasta proprietate este initializata automat cu prima componenta MainMenu asezata pe forma; in executie valoarea acestei proprietati poate fi modificata, printr-o instructiune de atribuire, astfel ca forma poate sa foloseasca mai multe componente MainMenu.

Proprietatea Name

Numele componentei. Orice componenta trebuie sa aiba un nume unic.

Proprietatea MenuItem

Se utilizeaza pentru a afla sau a preciza optiune dintr-un meniu care devine activa sau inactiva atunci cand un obiect OLE de pe formular este selectat sau deselectat.

Proprietatea ParentBidiMode(boolean)

Stabileste daca controlul foloseste aceeasi valoare pentru BidiMode ca si parintele.

Proprietatea ParentFont(boolean)

Stabileste daca controlul foloseste acelasi font ca si parintele

Proprietatea PixelPerInch

Se utilizeaza in timpul rularii programului pentru a modifica scara de marire a formei la rezolutia curenta. Daca valoarea este modificata fata de valoarea implicita, forma nu va avea aceleasi proportii la diferite rezolutii. Pentru a avea efect trebuie ca proprietatea Scaled sa fie true.

Proprietatea PopupMenu (de tip TpopupMenu)

Identifica componenta de meniu flotant care este utilizata de forma la un moment dat.

Se mentioneaza faptul ca aceasta proprietate nu este initializata automat cu componenta PopupMenu asezata pe forma. In executie valoarea acestei proprietati poate fi modificata, printr-o instructiune de atribuire, astfel ca forma poate sa foloseasca mai multe componente PopupMenu.

Proprietatea Position

Se refera la dimensiunea formei si al locul de plasare a acesteia la executie, fata de locul si dimensiunea care au fost utilizate in momentul realizarii proiectului. Valorile posibile sunt:

Valoare

Semnificatie

poDesigned

Forma va avea aceeasi pozitie si dimensiuni ca in momentul elaborarii proiectului.

poDefault

Pozitia si dimensiunea ferestrei vor fi stabilite de catre Windows

poDefaultPosOnly

Pozitia este stabilita de catre Windows, se pastreaza dimensiunile.

poDefaultSizeOnly

Pozitia este pastrata, Windows-ul alege dimensiunile.

poScreenCenter

Dimensiunile ferestrei se pastreaza, plasandu-se in centrul ecranului. Se fac ajustari pentru aplicatiile multimonitor.

po

Dimensiunile ferestrei se pastreaza, plasandu-se in centrul ecranului. Nu se fac ajustari pentru aplicatiile multimonitor.

PoMainFormCenter Dimensiunile ferestrei se pastreaza, plasandu-se in centrul ferestrei principale a aplicatiei. Nu se fac ajustari pentru aplicatiile multimonitor.

PoOwnerFormCenter Dimensiunile ferestrei se pastreaza, plasandu-se in centrul ferestrei specificate de proprietatea Owner.

Proprietatea PrintScale

Se utilizeaza pentru a afla sau stabili proportiile ferestrei tiparite. Valorile posibile sunt: Valoare

Semnificatie

poNone

Nu se scaleaza deloc.

poProportional

Fereastra este scalata astfel incat imaginea tiparita are aproximativ aceeasi dimensiune ca si pe ecran.

poPrintToFit

Forma este tiparita folosind aceleasi proportii ca pe ecran, dar modificate astfel incat sa incapa pe pagina.

Proprietatea Scaled

Specifica daca forma va fi redimensionata in functie de valoarea PixelsPerInch

Proprietatea ShowHint

Daca are valoarea true atunci va fi afisata o scurta informatie (Hint) despre obiectul deasupra caruia s-a oprit mouse-ul.

Proprietatea Tag

Utilizata pentru memorarea unei valori intregi. Nu are un rol predestinat.

Proprietatea UseDockManager

Stabileste daca se utilizeaza un manager pentru operatia de drag-and-dock.

Proprietatea Visible

Daca este true atunci componenta este vizibila.

Proprietatea WindowMenu

Se foloseste pentru a alfa sau stabili ce meniu se utilizeaza pentru o fereastră parinte in cazul aplicatiilor MDI (multiple document interface).

Proprietatea WindowState

Se refera la modul de afisare pe ecran a ferestrei. Poate avea urmatoarele valori

Valori

Semnificatie

wsNormal

Fereastra este afisata normal.

wsMinimized

Fereastra este afisata miminizata

wsMaximized

Fereastra este afisata maximizata.

Metode

Metoda BringToFront Aduce forma in fata.

Metoda Close

Se utilizeaza pentru inchiderea formei.

Metoda Hide Ascunde forma.

Metoda Refresh, Repaint Redeseneaza forma.

Metoda ScaledBy(M, D:integer)

Scaleaza toate obiectele de pe forma de M/D ori. Spre exemplu daca M=15 si D=7 dimensiunile tuturor componentelor se vor inmulti cu valoarea 15/7, dimensiunile fonturilor vor fi modificate de asemenea.

Metoda SendToBack

Trimite forma in spatele tuturor ferestrelor existente pe ecran.

Metoda SetFocus

Se utilizeaza pentru a selecta obiectul (controlul).

Metoda Show

Face vizibila forma.

Evenimente

Pentru fiecare dintre evenimentele prezentate mai jos Delphi-ul ne permite sa scriem rutina care sa trateze evenimentul respectiv. Daca in Object Inspector executam dublu clic pe

unul dintre evenimente Delphi-ul va deschide editorul de cod in care a scris deja antetul procedurii care va trata evenimentul, noi urmand sa scriem partea de cod care va trata evenimentul. Procedura de tratare a evenimentului va avea cel putin un parametru `Sender:TObject` care ne va indica cine a generat evenimentul, parametru care in general nu e folosit. Daca mai sunt necesari anumiti parametri pentru a trata evenimentul atunci acestia vor fi inclusi, spre exemplu pentru evenimentul `OnKeyPress`, care apare la apasarea unei taste, va fi transmis ca parametru si tasta(caracterul) care a fost apasata.

Evenimentul `OnActivate`

Apare atunci cand forma devine activa.

Evenimentul `OnCanResize`

Apare atunci cand se redimensioneaza controlul. Avem posibilitatea sa stabilim noile dimensiuni (`NewWidth`, `NewHeight :Integer`) si sa stabilim daca suntem de acord cu redimensionarea (`Resize:Boolean`)

Evenimentul `OnClick`

Apare atunci cand utilizatorul executa clic pe forma.

Evenimentul `OnClose`

Apare la inchiderea formei. Modificand parametrul `Action` avem posibilitatea de a decide ce se va executa. Parametrul `Action` poate avea valorile: Valoare Descriere
`caNone` Forma nu are voie sa se inchida, deci nu se intampla nimic.

`caHideForma` va fi ascunsa. Aplicatia mai poate accesa.

`caFree` Forma este inchisa, iar memoria alocata ei va fi eliberata

`caMinimize` Forma va fi minimizata.

Evenimentul `OnCloseQuery` Apare cand metoda `Close` a fost apelata. Prin modificarea parametrului

`CanClose:Boolean` decidem daca forma se poate inchide sau nu.

Evenimentul `OnConstrainedResize`

Apare imediat dupa evenimentul `OnCanResize`. Se utilizeaza pentru aplicarea limitarilor dimensiunilor, parametrii `MinWidth`, `MinHeight`, `MaxWidth`, `MaxHeight` stabilind lungimea minima, inaltimea minima, lungimea maxima, inaltimea maxima. La iesirea din procedura de tratare a evenimentului vor fi aplicate constrangerile si se vor obtine noile dimensiuni, dupa care va apare evenimentul `OnResize` care va permite ajustarile finale.

Evenimentul `OnContextPopup`

Apare la clic dreapta. Parametrul `MousePos:Tpoint` ne va indica pozitia mouse-ului, iar valoare pe care vom atribui parametrului `Handle:Boolean` va decide daca va fi afisat

meniul(daca `Handle=false`, si proprietatea `AutoPopup=true`) sau daca nu se va executa nimic (`Handle=true`).

Evenimentul `OnCreate` Apare la crearea formei.

Evenimentul `OnDblClick`

Apare atunci cand utilizatorul executa dublu clic pe forma.

Evenimentul OnDeactivate

Apare cand forma pierde focusul.

Evenimentul OnDestroy

Apare cand forma este distrusa.

Evenimentul OnDockDrop

Apare cand un obiect este depus pe forma prin operatia de drag-and-dock. Ca parametri vom avea Source care ne va indica de unde a fost luat obiectul, si pozitia la care a fost lasat X, Y.

Evenimentul OnDockOver

Apare cand un obiect este tras peste forma pentru operatie de drag-and-dock. Ca parametri vom avea Source care ne va indica de unde a fost luat obiectul, pozitia la care se afla X, Y si Accept care daca are valoarea true se accepta operatia, altfel nu.

Evenimentul OnDragDrop

Apare cand utilizatorul da drumul obiectului adus prin operatia de drag-and-drop. Ca parametri vom avea Source care ne va indica de unde a fost luat obiectul, si pozitia la care a fost lasat X, Y.

Evenimentul OnDragOver

Apare cand un obiect este tras peste forma pentru operatie de drag-and-drop. Ca parametri vom avea Source care ne va indica de unde a fost luat obiectul, pozitia la care se afla X, Y si Accept care daca are valoarea true se accepta operatia, altfel nu.

Evenimentul OnEndDock

Apare cand operatia de tragere s-a terminat. Se utilizeaza pentru a trata situatia in care operatia de drag-and-dock s-a terminat.

Evenimentul OnGetSiteInfo

Apare inainte de OnDockDrop daca proprietate DockSite este true.

Evenimentul OnHelp

Apare cand forma primeste cerere de ajutor (Help).

Evenimentul OnHide

Apare atunci cand forma este ascunsa (spre exemplu daca Visible ia valoarea false)

Evenimentul OnKeyDown

Apare la apasarea unei taste, daca componenta (in acest caz forma) este selectata. Parametrul Key ne da codul tastei apasate, iar parametrul Shift, care poate avea valorile ssShift, ssAlt, ssCtrl, ne indica care dintre tastele Shift, Ctrl sau Alt a mai fost apasata odata cu tasta. Pentru tastele care nu sunt litere sau cifre se pot folosi codurile virtuale (ex. VK_PAUSE, VK_RETURN, VK_ESCAPE ...)

Evenimentul OnKeyPress

Apare daca utilizatorul apasa tasta care genereaza un singur caracter. Parametrul Key contine caracterul tastei apasate. Daca a fost apasata combinatie de taste, spre ex. Shift+F1, nu provoaca acest eveniment.

Evenimentul OnKeyUp

Apare la eliberarea unei taste, daca componenta (in acest caz forma) este selectata. Parametrul Key ne da codul tastei apasate, iar parametrul Shift ne indica care dintre tastele Shift, Ctrl sau Alt a mai fost apasata odata cu tasta. Pentru tastele care nu sunt litere sau cifre se pot folosi codurile virtuale (ex. VK_PAUSE, VK_RETURN, VK_ESCAPE ...)

Evenimentul OnMouseDown

Apare la apasarea unui buton al mouse-ului. Parametrul Button poate avea valorile mbLeft, mbRight, mbMiddle si ne indica care dintre butoanele mouse-ului au fost apasate. Parametrul Shift ne indica care dintre butoanele Ctrl, Shift sau Alt au fost apasate, iar parametrii X, Y ne indica pozitia mouse-ului.

Evenimentul OnMouseMove

Apare la miscarea mouse-ului peste un control(in acest caz forma). Parametrul Shift ne indica care dintre tastele Shift, Alt, Ctrl au fost apasate, iar parametrii X, Y ne indica pozitia mouse-ului.

Evenimentul OnMouseUp

Apare la eliberarea unui buton al mouse-ului. Parametrul Button poate avea valorile mbLeft, mbRight, mbMiddle si ne indica care dintre butoanele mouse-ului au fost apasate. Parametrul Shift ne indica care dintre butoanele Ctrl, Shift sau Alt au fost apasate, iar parametrii X, Y ne indica pozitia mouse-ului.

Evenimentul OnMouseWheel

Apare cand rotita mouse-ului este rotita, daca evenimentul nu este tratat (parametrul Handle are valoare False) atunci apare evenimentul OnMouseWheelUp sau evenimentul OnMouseWheelDown, in functie de directia in care este invaritata rotita. Parametrul Shift ne indica care dintre tastele Ctrl, Alt sau Shift au fost apasate, parametrul WheelDelta ne indica de cate ori a fost invaritata rotita.

Evenimentul OnMouseWheelDown

Apare la invartitura rotitei in jos, daca evenimentul OnMouseWheel nu a fost tratat.

Procedura are aceesi parametri ca si cea ce trateaza evenimentul OnMouseWheel

Evenimentul OnMouseWheelUp

Apare la invartitura rotitei in sus, daca evenimentul OnMouseWheel nu a fost tratat.

Procedura are aceesi parametri ca si cea ce trateaza evenimentul OnMouseWheel

Evenimentul OnPaint

Apare atunci cand forma este redesenata.

Evenimentul OnResize

Apare la redimensionarea formei.

Evenimentul OnShortCut

Apare la apasarea unei taste inainte de evenimentul OnKeyDown. Se utilizeaza pentru a trata short-cut-urile inainte de a trata apasarile obisnuite de taste.

Evenimentul OnShow

Apare atunci cand forma este facuta vizibila.

Evenimentul OnStartDock

Apare atunci cand utilizatorul incepe sa traga un obiect peste control, iar proprietatea DragKind are valoarea dkDock.

Evenimentul OnUnDock

Apare atunci cand utilizatorul incearca sa ia un obiect ce se afla pe forma. Parametrul Client ne indica obiectul ce va fi luat, parametrul NewTarget ne indica unde va fi dus obiectul, iar parametrul Allow il folosim pentru a permite operatia (daca ii atribuim true), sau aointerzice.

Probleme si intrebari

1. Cum putem face invizibile butoanele sistem?
2. Cum putem realiza o forma de dimensiune 100x100 care sa nu poata fi redimensionata?
3. Scrietioaplicatie care sa afiseze pozitia mouse-ului ca titlu a formei.
4. Scrietioaplicatie care sa se inchida daca se apasa butonul stang al mouse-ului pe forma.

5. Scrietioaplicatie care sa aibaforma(fereastră) care sa "fuga" de mouse-ul utilizatorului ori de cate ori acesta ajunge cu mouse-ul pe ea.

4.3 Lectia 3 - Butonul

Butonul este des utilizat pentru a initia diverse actiuni la apasare. In Delphi exista trei tipuri de butoane:

Imagine	Tip buton	Descriere
---------	-----------	-----------

Button
butonul standard

BitBtn

denumirea vine de la Bitmap Button si permite asezarea unui bitmap (imagini) in interiorul butonului. Exista posibilitatea de a stabili cateoimagine pentru fiecare stare a butonului: ridicat, apasat, dezactivat.

SpeedButton

permite adaugarea unei (a unor) imagini care poate (pot) acoperi intreg butonul. Butoanele de acest tip se pot grupa.

Deoareceomare parte din proprietatile, metodele si evenimentele butoanelor sunt comune cu cele de la form voi trata doar ceea ce apare nou.

4.3.1 Button(TButton)

Se utilizeaza pentru a pune un buton standard pe form.

Proprietatile butonului

Proprietatea Action

Este utilizata pentru centralizarea raspunsurilor la comenzile date de utilizator.

Majoritatii componentelor li se poate asocioactiune, actiunile fiind centralizate si gestionate folosind componenta ActionList din pagina standard.

Proprietatea Anchors

Se utilizeaza pentru a ne asigura ca obiectul isi pastreaza pozitia relativ la marginile parintelui(a componentei careocontine), chiar daca componenta parinte este redimensionata.

Cand parintele este redimensionat obiectul isi pastreaza pozitia relativ la marginile de care a fost ancorat.

Proprietatea BidiMode

Permite ajustarea modului de afisare si a comportamentului automat atunci cand aplicatia ruleaza intr-ozona in care se citeste de la dreapta la stanga si nu de la stanga la dreapta.

Proprietatea Cancel(boolean)

Daca are valoarea true atunci la apasarea butonului Escape va avea loc evenimentul

OnClick

Proprietatea Capiton (de tip string)

Specifica un sir de caractere care este afisat pe buton, de exemplu Button1. In cazul componentelor aceasta proprietate desemneaza eticheta atasata componentei. In acest ultim

caz litera precedata de caracterul "&" este litera de selectie a componentei (litera de selectie este subliniata si componenta poate fi selectata prin combinatia Alt - litera subliniata).

Proprietatea Constraints

Se utilizeaza pentru a stabili dimensiunea minima si maxima pentru componenta.

Aceste dimensiuni nu pot fi depasite.

Proprietatea Cursor

Stabileste imaginea utilizata pentru reprezentarea cursorului mouse-ului.

Pentru selectarea unei imagini se vor folosi diferitele constante predefinite de descriere a cursorului. Valoarea implicita este constanta crDefault.

Proprietatea Default

Daca are valoarea true atunci la apasarea tastei Enter va avea loc evenimentul OnClick.

Proprietatea DragCursor

Specifica ce imagine sa fie utilizata pentru buton atunci cand acesta este tras (drag)

Proprietatea DragKind

Se foloseste pentru a stabili daca se iau obiecte de pe control(drag) sau se pot pune pe

el(dock).

Proprietatea DragMode

Stabileste daca utilizatorul poate executa operatia de drag(tragere). Daca valoarea

proprietatii este dnManual utilizatorul nu poate executa operatia, iar daca valoarea proprietatii este dmAutomatic atunci poate executa operatia de drag.

Proprietatea Enabled (de tip boolean)

Se refera la accesibilitatea componentei. Daca se alege varianta True, care este si valoarea implicita, atunci componenta va reactiona la diferitele evenimente referitoare la mouse, tastatura si ceas. Daca se alege varianta False, componenta va fi inhibata si nu va raspunde la evenimentele echipamentelor de intrare. Valoarea acestei proprietati de regula este modificata in executie, de exemplu

pentru a interzice utilizarea unor butoane de comanda in contextul respectiv. Eticheta unei componente inhibate este afisata cu caractere estompate. O componenta inhibata ulterior poate fi reactivata, moment in care eticheta componentei va fi afisata cu caractere negre.

Proprietatea compusa Font

Stabileste fontul utilizat la textele care sunt afisate pe obiectul Canvas al formei

precum si la textele aferente componentelor asezate pe forma. Fontul utilizat poate fi definit in perioada de elaborare a proiectului, insa poate fi modificat in executie, prin redefinirea subproprietatilor Color (culoare), Height (inaltime), Name (nume),

Pitch (latimea caracterelor), Size (dimensiune), sau Style (stil).

Proprietatile Height si Width (de tip integer)

Sabilesc inaltimea, respectiv latimea formei sau componentei. Aceste valori sunt exprimate in puncte care include si dimensiunile rezervate chenarelor.

Proprietatea HelpContext

Se foloseste pentru a stabili un numar folosit in apelarea ajutorului(Help) context-senzitiv, pentru a determina care fereastră de ajutor este afisata daca utilizatorul apasa tasta F1. Daca HelpContext are valoarea 0 atunci va fi afisata fereastră de ajutor corespunzatoare parintelui.

Proprietatea ModalResult

Proprietatea se poate utiliza pentru a inchide fereastră modală (care a fost făcută vizibilă cu metoda ShowModal), returnând valoarea din ModalResult în câmpul ModalResult

al ferestrei modale. Nu este necesar sa scriem cod care sa trateze apasarea butonului. Pentru clarificare vezi Exemplul 1.

Proprietatea	poate avea mai multe valori:	Constanta	Valoare	Ce face
mrNone	0	Nimic		

MrOk idOK Utilizatorul a iesit apasand butonul OK

mrCancel idCancel Utilizatorul a iesit apasand butonul CANCEL

mrAbort idAbort Utilizatorul a iesit apasand butonul ABORT

mrRetry idRetry Utilizatorul a iesit apasand butonul RETRY

mrIgnore idIgnore Utilizatorul a iesit apasand butonul IGNORE

MrYes idYes Utilizatorul a iesit apasand butonul YES

MrNo idNo Utilizatorul a iesit apasand butonul NO

Proprietatea ParentShowHint(boolean)

Se utilizeaza pentru a ne asigura ca fie toate componentele afiseaza Hint-ul fie toate nu il afiseaza. Daca ParentShowHint e true atunci obiectul foloseste ShowHint-ul parintelui, altfel decizia de a afisa Hint-ul va fi luata in functie de ShowHint-ul propriu.

Proprietatea TabOrder

Poate lua valori in intervalul -1..32767 si stabileste la a cata apasare a tastei Tab se ajunge la obiectul(controlul) respectiv. Primul obiect adaugat are valoarea proprietatii TabOrder 0, urmatorul 1, al treilea 2 s.a.m.d.

Proprietatea TabStop(boolean)

Se foloseste pentru a permite sau interzice accesul la control prin folosirea tastei Tab.

Metode

Metoda Click

Simuleaza apasarea butonului.

Evenimente

Evenimentul OnEnter

Apare atunci cand controlul(in cazul nostru butonul) primeste focus-ul, spre exemplu cand se ajunge cu Tab pe el.

Evenimentul OnExit

Apare atunci cand controlul pierde focus-ul, spre exemplu cand un alt control de pe fereastra primeste focus-ul. Pentru lamuriri vezi exemplul 2.

Exemple

Exemplul 1

Urmatoarea aplicatie va folosi proprietatea ModalResult a butonului.

Creationoua aplicatie care are doua form-uri(ferestre): Form1 si Form2.

Pe Form1 puneti un buton a carui eveniment OnClick e tratat de procedura:

```
procedure TForm1.Button1Click(Sender: TObject); begin
```

```
if form2.ShowModal=mrOk then
```

```
form1.Caption:='OK'
```

```
else form1.Caption:='No';
```

```
end;
```

Pe Form2 puneti doua butoane, unul avand

ModalResult=mrOk, iar celalaltavand

ModalResult=mrNo.

Compilati, lansati in executie si testati functionarea programului apasand succesiv pe cele doua butoane din fereastra a doua(Form2).

Exemplul 2

Pentru a ilustra folosirea evenimentelor OnEnter si OnExit creamo o aplicatie pe a carei fereastra puneti doua butoane: Button1 si Button2, iar pentru Button2 scrieti codul urmatoare pentru tratarea evenimentelor OnEnter si OnExit:

```
procedure TForm1.Button2Enter(Sender: TObject); begin
```

```
button2.Caption:='Enter'; end;
```

```
procedure TForm1.Button2Exit(Sender: TObject); begin
```

```
button2.Caption:='Exit'; end;
```

Compilati programul, lansati-l in executie dupa care treceti de la un buton la altul folosind tasta Tab.

4.3.2 BitBtn(TBitBtn)

Are aceleasi proprietati, metode si evenimente ca si butonul clasic, in plus adaugandu-se posibilitatea de a avea o imagine(bitmap) pe buton.

Proprietati

Proprietatea Glyph

Care stabileste ce imagine va fi afisata pe buton. Putem avea maxim patru imagini in fisierul bitmap, toate imaginile fiind de aceeaasi dimensiune, asezate pe aceeaasi linie una langa cealalta. Pe buton va fi afisata una dintre imagini in functie de starea butonului.

Pozitia imaginii

Starea butonului

Descriere

Prima

Sus

Aceasta imagine apare daca butonul e in pozitia ridicat. Daca nu exista alte imagini atunci aceasta imagine este utilizata si pentru celelalte stari.

A doua

Dezactivat

Apare daca butonul nu poate fi selectat (proprietatea enable = false)

A treia

Apasat

Apare cand s-a facut clic pe buton.

A patra

Jos

Apare daca butonul a ramas jos.

Proprietate Kind

Stabileste ce imagine se va afisa pe buton, precum si valoarea ModalResult. Poate avea urmatoarele valori: Valoare Buton Descriere

bkCustom Va fi afisata imaginea din Glyph, valoarea ModalResult trebuie stabilita manual

Se poate utiliza pentru anulara unei actiuni.

bkAbort

bkAll Este recomandat a se utiliza pentru selectia a mai multor optiuni.

bkCancel Valoarea proprietatii Cancel este setata automat la valoarea true, ModalResult va avea valoarea mrCancel

La apasarea butonului fereastra se va inchide. Valoarea

bkClose

proprietatii Default este true.

La apasarea butonului va fi apelat Help-ul programului.

bkHelp

Este recomandat a se utiliza daca a aparut o eroare iar

bkIgnore

utilizatorul doreste sa continue operatia

bkNo Valoarea proprietatii Cancel este setata automat la valoarea true, ModalResult va avea valoarea mrNo

bkOk Valoarea proprietatii Default este pusa automat pe valoarea true, ModalResult va avea valoarea mrOk

Este recomandat a se utiliza daca a aparut o eroare iar

bkRetry

utilizatorul doreste sa reincearca operatia.

bkYes Valoarea proprietatii Default este pusa automat pe valoarea true, ModalResult va avea valoarea mrOk

Proprietatea Layout

Stabileste pozitia imaginii pe buton. Poate avea urmatoarele valori:

Valoare

Descriere

blGlyphLeft

Pune imaginea in partea stanga a textului.

blGlyphRight

Pune imaginea in partea dreapta a textului.

blGlyphTop

Pune imaginea deasupra textului.

blGlyphBottom

Pune imaginea sub text.

Proprietatea Margin

Stabileste distanta, in pixeli, intre imagine si marginea butonului. Daca valoarea este - 1 (valoare implicita) atunci imaginea si textul vor fi centrate

Proprietatea NumGlyphs

Indica numarul de imagini care se afla in bitmap-ul memorat in Glyph. Numarul de imagini poate fi intre 1(valoare implicita) si 4.

Proprietatea Spacing

Stabileste numarul de pixeli dintre imagine si text, implicit are valoarea 4. Daca are valoarea -1 atunci textul va fi centrat intre imagine si marginea butonului.

Proprietatea Style

Determina modul in care este afisata imaginea pe buton. Poate avea urmatoarele valori: Valoare

Descriere

bsAutoDetect

Daca folosim Windows 3.1 bitmap-ul va avea infatisarea din Windows 3.1, altfel va avea noua infatisare.

bsWin31

Va folosi infatisarea standard din Windows 3.1

bsNew

Va folosi noua infatisare

4.3.3 SpeedButton(TSpeedButton)

Fata de butonul clasic TSpeedButton-ul introduce proprietati care sa-i permita lucrul in grupuri de butoane, spre exemplu in bare de butoane.

Proprietati

Proprietatea AllowAllUp

Determina daca toate butoanele ce fac parte dintr-un grup pot sa fie toate ridicate simultan sau nu. Daca are valoare true atunci butoanele ce fac parte din acelasi grup pot fi toate ridicate, altfel unul dintre butoane va fi apasat, iar la apasarea unui alt buton butonul apasat se va ridica. Daca campul GroupIndex este 0 atunci AllowAllUp nu are nici un efect.

Proprietatea Down

Se foloseste pentru a afla daca butonul este apasat sau nu. Proprietatea este valabila doar daca proprietatea GroupIndex este diferita de 0.

Proprietatea Flat

Daca are valoarea true butonul nu va fi afisat 3D. Daca proprietatea are valoarea true putem folosi imagini(bitmap) care sa ilustreze starea butonului.

Proprietatea GroupIndex

Permite sa selectam din ce grup face parte butonul. Daca are valoarea 0 atunci fiecare buton actioneaza separat, iar daca are o valoare mai mare ca 0 atunci desemneaza numarul grupului din care face parte butonul. Daca utilizatorul apasa un buton din grup acesta ramane apasat pana cand un alt buton din grup va fi apasat.

Proprietatea Transparent

Specifica daca fundalul butonului este transparent sau nu.

Intrebari si probleme

1. Cum procedam ca un buton sa nu poata fi selectat decat cu mouse-ul?
2. Sa se scrie o aplicatie care afiseaza o fereastra pe care se afla un buton care "fuge" ori de cate ori utilizatorul a ajuns cu mouse-ul pe el, dar nu poate iesi in afara ferestrei.
3. Sa se scrie o aplicatie care afiseaza o fereastra pe care se afla trei butoane din care unul singur poate fi selectat la un moment dat si care afiseaza un mesaj, ca titlu al ferestrei, in functie de ce buton este apasat.
4. Sa se scrie o aplicatie care are butoane ce au alta culoare decat a ferestrei.

4.4 Lectia 4 -Edit

Se utilizeaza pentru a permite programului sa preia un text de la utilizator. Se poate utiliza si pentru a afisa un text.

Proprietati

Proprietatea AutoFocus

Daca are valoarea true atunci textul va fi selectat cand controlul va primi focus-ul. Se pune pe valoarea true daca e mai probabil ca utilizatorul sa inlocuiasca textul existent decat sa adauge ceva la el.

Proprietatea AutoSize

Daca are valoarea true atunci controlul isi va modifica inaltimea daca inaltimea textului se schimba, daca nu exista posibilitatea ca textul sa fie trunchiat.

Proprietatea BorderStyle

Determina daca controlul va avea o bordura, si atunci are valoarea bsSingle, sau nu, si atunci are valoarea bsNone.

Proprietatea CharCase

Determina ce fel de litere se pot folosi(mari, mici sau combinat). Proprietatea poate lua urmatoarele valori: Valoare

Descriere

ecLowerCase

Literele sunt transformate in litere mici

ecNormal

Se pot utiliza atat litere mari cat si litere mici

ecUpperCase

Literele sunt transformate in litere mari

Proprietatea HideSelection

Determina daca textul ramane selectat atunci cand controlul pierde focusul. Daca are valoare false atunci la pierderea focusului textul ramane selectat. Se poate accesa textul selectat prin intermediul proprietatii SelText. Pozitia de inceput a selectiei este memorata in proprietatea SelStart, iar lungimea textului selectat e memorata in SelLength.

Proprietatile ImeMode si ImeName

Se utilizeaza pentru a stabili anumite optiuni pentru editorul IME ce se utilizeaza pentru limbile asiatice.

Proprietatea MaxLength

Stabileste numarul maxim de caractere ce poate fi introdus. Daca are valoarea 0 atunci nu exista nici un fel de limitare.

Proprietatea OEMConvert

Daca are valoare true atunci caracterelor vor fi "traduse" din ANSI in OEM si invers pentru a elimina caracterele ce nu pot fi folosite la denumirea unui fisier.

Proprietatile ParentBiDiMode, ParentColor, ParentCtl3D, ParentFont, ParentShowHint

Au valoarea true daca se doreste ca proprietatea a carei nume apare dupa Parent sa aiba aceeasi valoare ca si a parintelui(cu a componentei pe care se afla asezat)

Proprietatea PasswordChar

Specifica ce caracter se va afisa daca se doreste citirea unei parole care nu trebuie sa apara pe ecran. Daca are valoarea #0 atunci caracterele se vor afisa normal.

Proprietatea ReadOnly

Daca are valoarea true nu se va putea modifica continutul Edit-ului.

Proprietatea Text

Contine textul ce se afiseaza in Edit. Proprietatea poate fi utilizata pentru a prelua ceea ce a tastat utilizatorul.