

ДӘРІС 14.ПРОГРАММА ИНТЕРФЕЙСІН ҚҰРУ.  
АСПАПТАРДЫ ӨҢДЕУ ПРИНЦИПТАРЫ.  
ИНТЕРФЕЙС ҚҰРУ ӘДІСТЕРІ ЖӘНЕ  
АСПАПТЫҚ ҚҰРАЛДАРЫ

---



**Мақсаты:**

**Интерфейс құру әдістері және аспаптық құралдарын білу.**

- Әрекет (Action) деп операцияны атаймыз, оны пайдаланушы интерфейс элементтеріне әрекеттесіп жасайды. Әрекет орындалу керек компонент әрекет (Action target) мақсаты деп аталады. Әрекет инициализацияланған компонент (батырма, пункт меню), — әрекет клиенті (Action client) деп аталады.
- Ескерту Барлық әрекеттер бір компоненттер TActionList немесе TActionManager тізіміне біріккен кезде ғана жұмыс істей алады. Бұл компоненттердің сыртынан ешбір әрекет орындалмайды.
- Бірінші жұмыс формаға TActionList компонентін орнатудан басталады. Ол бірінші **Standard** бетіндегі компоненттер палитрасында орналасқан. Бұдан кейін әрекеттер тізімінің редакторын маустың басқышын екі рет компонент немесе контекстік менюді басу арқылы жіберу керек.
- Әрекеттермен байланысты уақиғалар TAction компоненті үш уақиғаға көңіл бөледі: OnExecute, OnUpdate И OnHint.

- Бірінші — және бұл негізгі — берілген әрекетке берілетін реакция. Бұл уақиға басқышты басу кезінде орындалады. Мұнда — тәртіпке байланысты орындалады. Тәртіпке байланысты дер отырғанымыз, ол сигналды өңдеу сұлбасы бойынша орындалады, оның 4 - этапы бар:
- 1. Бірінші әрекеттер тізімінен OnExecute уақиға өңдегіші шақырылады
- TActionList:
- property OnExecute: TActionEvent; TActionEvent = procedure (Action: TBasicAction; var Handled: Boolean)
- of object;
- Егер бұл уақиғаның өңдегіші сізбен қаралмаса онда келесі уақиға генерациясы орындалады — қадам 2.
- 2. Глобальды Application объектісінің onActionExecute уақиғалар өңдегіші шақырылады (уақиға типі алдындағыдай — TActionEvent). Егер әрекет сигналын орындамаса онда келесі қадамға өтеміз.
- 3. Әрекеттің өзінен onExecute уақиға өңдегіші шақырылады (объекттипі TAction немесе мұраға алынады).
- 4. Егер бірінші үш қадам жағдайды өндемесе (False болса), онда, мүмкін, бұл дұрыс қойылмаған мақсатқа байланысты болған шығар (Target). Соңғы шанс ретінде қосымшаға CM\_ACTIONEXECUTE хабарламасы жіберіледі. Бұл жағдайда бұл жағдай үшін басқа мақсат ізделеді.

- onupdate уақиғаларды енгізу. onupdate уақиғаларды қолдануға мысал:
- procedure TForm1.PasteActionUpdate(Sender: TObject); begin
- TAction(Sender).Checked := Clipboard.HasFormat(CFJTEXT);
- end;
- **Ескерту** onupdate уақиғаларды шақыру кезінде де 4 – этаптық әрекеттер тізбегі орындалады, ол алдыңғы OnExecute –ге ұқсас.
- Үшінші уақиға типі келесідегідей:
- THintEvent = procedure (var HintStr: string; var CanShow: Boolean) of object;
- Ол басқару элементінен берілген әрекетке байланысты жауаптар керек болған кезде шақырылады. Уақиғалар өңдегішінде бірнәрсе шығатын шықпайтындығын көрсетуге болады (CanShow параметрі) және, егер шықса онда не шығатындығы (Hintstr параметрі).
- Бұлар Taction компонентіне байланысты уақиғалар болған. Ал TActionList компонентінің өзінің үш уақиғасы бар: OnExecute, OnUpdate және OnChange. Бірінші екеуін біз қарастырдық; ал үшінші тізімді өзгерту кезінде орындалады (әрекеттерді қосу немесе жою).

- **Әрекеттер клиенттеріне таратылатын қасиеттер** Егер бірнеше басқыштардың немесе меню пунктерінің бір ғана жалпы өңдегіші болса, онда олардың басқа қасиеттерінің де бірдей болуын талап еткен дұрыс. Delphi –де ол солай тартылған.
- **TActionManager** компоненті (ары қарай тексте — әрекеттер менеджері). Ол басқа да көптеген қосымша мүмкіндіктер береді.
- TActionManager редакторының бірінші бетінде (екілік маусты басу арқылы немесе контексті менюдегі **Customize** командасы арқылы шақырылады) барлық панелдер тізімі тұрады, олар берілген әрекеттер менеджерімен байланысқан. Сіз жаңа компонент қосып немесе бұрынғы компонентті TActionToolBar басқышындағы **New** және **Delete** басу арқылы орындай аласыз.
- Жүйенің стандартты интерфейсі. Интерфейсті өңдеу мысалы – MS Visual Systems, Borland Visual Systems, Oracle Visual Systems және т.б. Шешімі, артықшылықтары, кемшіліктері, мобильдігі, тереңдігі, түсі, өлшемі, анимациясы, масштабталуы, сенімділігі, қатынау жылдамдығы және т.б. Жүйенің стандартты интерфейсінің элементтері және олардың компоненттік таратылуы.

- Кірісу минимизациясы. Автоматизация деңгейі және компетенттік емес жағдайлардың сақтау. Әрекеттер тізбегі және сеанстары. Шаблондарды қолдану. Көптүрлілік және мобильдік көрсеткіштері. Жүйелерді қолдану, – аспаптарды, істік графиканы.
- Бір қосымшаның екі рет жіберілуінен қалай құтылуға болады?
- Мұндай жағдай көп кездеседі. Мұндағы негізгі идея, ол қосылатын қосымша қосылар алдында ресурсты алып алу керек, ал қалғандары да оны орындағысы келеді бірақ ол бос бюолмағандықтан өз жұмыстарын аяқтайды да оның басқа қосымшалары жасалмайды.
- Мұндай ресурстың мысалы — жадығы бейнеленетін файлдағы жалпы блок. Бұл ресурстың аты болғандықтан, оны өз қосымшыңызға уникальды етіп жзасауға болады:
- `var UniqueMapping : THandle;`
- `FirstWindow : THandle`
- `; begin`
- `UniqueMapping := CreateFileMapping($ffffffff,`
- `nil, PAGE_READONLY, 0, 32, 'MyMap');`
- `if UniqueMapping = 0 then`
- `begin ShowMessage(SysErrorMessage(GetLastError)); Halt; end`
- `else if GetLastError = ERROR_ALREADY_EXISTS then`
- `begin FirstWindow := FindWindowEx(0, 0, TfmMain.ClassName, nil);`
- `if FirstWindow < 0 then SetForegroundWindow(FirstWindow); Halt; end;`
- `// Application.Initialize қосымшасының басқа көшірмесі жоқ;`

- Бұндай жолдарды жоба формасын құрудан бұрын текстін басын орныластыру керек.
- Жадымен бірге қолданылатын блокжүйеілік бет файлында беріледі (бұл жайында бірінші параметр айтып тұр, ол -1 тең, функция сипаттамасын қара.
- CreateFileMapping). Оның аты — муMap. Егер блокты құру кезінде қате коды алынса ERROR\_ALREADY\_EXISTS, онда ол қосымшаның жұмыс істеп тұрған көшірмесінің бар екендігін білдіреді.
- Экранға орналастыру. Бір- және көпбеттік интерфейс. Модальды терезелер және фокус. Панелдер, фреймдер, ағаштар, графтар, желілер – уақиғалық парадигмалардың визуальды бейнеленуінің ақпараттық моделі. Концентрация және интерфейсстің таратылу жалпыламасы. Интерфейстің көпбетілігі және көппроцессорлығы. Бейнелерді өзгерту – программадағы әрекеттер. Интерфейс элементтерінің активациясы және фокустың қолданылуы. Форманың модальдығы – артықшылықтары және кемшіліктері



- Интерфейсті визуальды жобалауды қамтамасыздандыратын технологиялар. Бояу. Операциялық жүйенің графикалық интерфейсінің кітапханасы. Визуальды программа құрушылардың графикалық кітапханасы. Графикалық бағытталған жүйелер. Графикалық файлдардың форматы – графикалық ақпарат жеткізгіштер. Экспорт – графикалық форматтардың импорты. Микропроцессордың графикалық мүмкіндіктері – деректер типі, операциялар, видеожүйелер.
- Терезелердің ауысуы және оларды орналастыру тәртібі. Терезелік парадигма. Алғы план және фон. Терезелер жиыны. Жалпылама және бөлшек бейнелер. Оперативті активтендіру.
- Көмекші сөздерді ұйымдастыру. MSWord – көмекші сөздерді ұйымдастыруға мысал. АПК, интеллекті өсіу стимуляторы және үйртуші. Көмекші сөздер – контекстілігі, нақтылығы, көпбөліктілігі, визуальдылығы, сыйысушылығы, қалпына келтірілуі, динамикалығы (оперативтігі) тестіленуі.