

ДӘРІС 9. WIZARDS – ШЕБЕРІНІҢ (МАСТЕР)  
АВТОМАТТЫ ӨҢДЕУІ. ПРОГРАММАНЫҢ МӘТІНІНЕ  
ҚАЙТАЛАП ҚОЛДАНУҒА ДАЙЫН ӨҢДЕУДІҢ  
КІТАПХАНАСЫ. ӨЗІНДІК КОМПОНЕНТТЕРДІ ҚҰРУ  
ЖӘНЕ ОЛАРДЫҢ ИНСТРУМЕНТАЛДЫҚ  
ПРОГРАММАЛЫҚ ҚҰРАЛ ЖАБДЫҚТАРЫН  
ҚОЛДАНЫП ОЛАРДЫ БЕЗЕНДІРУ.

---

**Мақсаты:**

**Программа объектілерін байланыстыру.**

- Программаны орындау және жобалау кезіндегі объектілер қасиетін өзгерту және қатынау әдістемесі. Қасиеттер мәнінің тізімі. Қасиеттер, объект визуализациясын және тәртібін бейнелеу. TObject ( TPersistent ( TComponent ( TControl ( TGraphicControl, TWinControl)))) – компоненттер ағашы.
- MAKE.EXE – программаны қайталау процессі кезіндегі компиляция. MAKE жұмысы орындалатын және объектілік файлдар арасындағы ақпараттық және уақыттық байланыстарға негізделген. MAKE программа үшін синтаксис:
- MAKE [options...] [target[target]]
- options
- are MAKE options that control how MAKE works
- target
- is the name of the target listed in the makefile that you want to build
- Өңдеудің стандартты компоненттер жиыны. Графикалық компоненттер. Өңдеушіге берілетін компонент палитрасы – құрамы, палитралық бет қызметі, визуальды және визуальды емес компоненттер (көру қасиеттері), форма бойынша компоненттерді орын орнымен орналастыру және көрсетуді өзгерту (визуализация қасиеттері). Графикамен жұмыс кезінде екі термин қолданылады - drawing and painting. Drawing элементар графикалық объектілер құру кезінде қолданылады (сызықтар, фигуралар) кодпен беріледі. Ол үшін сурет панелі қолданылады - палитра (object canvas) және объекта canvas әдістері. Painting – түсінігі, элементар графикалық объектілермен манипуляциялау кезінде қолданылады. Келесі тізім графикамен жұмысты оңайлату үшін қолданылады.

- Refreshing the screen (экранды қайта салу)
- Types of graphic objects (объект типтері)
- Common properties and methods of canvases (жалпы қасиеттер және палитра әдістері)
- Handling multiple drawing objects in an application (қосымшада қолмен салу)
- Drawing on a bitmap (биттік аймақта сурет салу)
- Loading and saving graphics files (графикалық файлдарды сақтау және жүктемеу)
- Using the clipboard with graphics (графика үшін буфер қолдану)
- Rubber banding example (мысал)
- Компоненттерді және компонентер пакетін құру, оларды қолдану. Компоненттерді визуальды жобалау әдістемесі. Өз компоненттеріңізді өңдеу және қосу. Жаңа компоненттерді екі түрлі әдіспен құруға болады – қолмен және компонент мастерінің көмегімен (Component wizard).

## ● Өңдеудің негізгі қадамдары:

- 1 Create a unit for the new component. (Жаңа компонент үшін модуль құру)
- 2 Derive your component from an existing component type. (Мұрагер типін анықтау)
- 3 Add properties, methods, and events. (әдістер, уақиғалар, қасиеттер қосу).
- 4 Register your component with the IDE. (компонентрегистрациясы)
- 5 Create a bitmap for the component. (ярлык құру)
- 6 Create a package (a special dynamic-link library) so that you can install your component in the IDE.
- 7 Create a Help file for your component and its properties, methods, and events. (көмек құру).
- Жұмыс аяқталғанда сіз келесі файлдарды аласыз:
- A package (.BPL) or package collection (.DPC) file (пакетанемесеколлекция пакеті)
- A compiled package (.DCP) file (компиляцияфайлы)
- A compiled unit (.DCU) file (объектілік файл)
- A palette bitmap (.DCR) file (ярлыкфайлы)
- A Help (.HLP) file (көмек файлы)
- Open Tools API интерфейстері. Мастерлер құру. Tools API аспаптарының барлық функциялары – ToolsAPI бір модульінде берілген.
- C++Builder және Delphi мастерлері, көп аспаптар үшін платформалары сәйкес болады, яғни wizard in Delphi құрып, оны C++Builder қолдануға болады.

- Мастер ToolsAPI-мен берілген сервистерді қолданады. Әрбір сервис – функциялар тобынан тұратын интерфейсті құрайды.
- Сервистер және басқа интерфейстер екі категорияға бөлінген, ол префикстерде белгіленген - NTA және OTA. NTA (native tools API) бар IDE объектілерге тікелей қатынауды қамтамасыздандырады, мысалы, TmainMenu сияқты. OTA (open tools API) пакеттеуді керек етпейді және IDE қатынау интерфейс арқылы орындалады.
- Tools API екі түрлі интерфейсті туындайды, біріншісі программист дайындайды, ал екіншісін IDE дайындайды. Сіз жасап шығаратын мұрагер интерфейстер үш категорияға бөлінеді: мастер (wizards), сипаттаушы (notifiers), құрушы (конструкторлар, creators):
- Сипаттаушы (*notifier*) – интерфейстің басқа түрі, оны IDE артқа қайтуды анықтау үшін қолданады. Құрушы (*creator*) интерфейстің басқа түрі, оны өзгерту керек. Жаңа модулдер құру үшін қолданылады.

- **Программа объектілерін байланыстыру.** Есеп берулер құру. Менеджерлер, редакторлар, мастерлер (wizards). Компонент құру мастері және компоненттері - Component wizard - құру үшін керек деректер: 1) мұрагер классты анықтау (The class from which the component is derived.); 2) жаңа классқа жаңа ат беру (The class name for the new component.); 3) палетта компонентінің бетін анықтау, онда жаңа класс орналасады (The Component palette page where you want it to appear.); 4) компонент орналасатын модуль атын анықтау (The name of the unit in which the component is created.); 5) модульге барар жолды көрсету – толық аты (The search path where the unit is found.); 6) жаңа компонент орналасқан пакет атын анықтау (The name of the package in which you want to place the component).