

**ДӘРІС 7.ВИЗУАЛЬДЫ ПРОГРАММАЛАУ ҚҰРАЛДАРЫ
– RATIONAL ROSE- DELPHI, C++, JAVA ОБЪЕКТІЛІ
БАҒДАРЛАМАЛЫҚ ПРОГРАММАЛАУ ТІЛДЕРІН
ВИЗУАЛДЫ ЖОБАЛАУ ИНСТРУМЕНТІ.**

Мақсаты:

Delphi, c++, java объектілі бағдарламалық программалау тілдерін визуалды жобалау инструментің игеру.

- Программаның визуальды жобалануы. Визуальды орталар (Delphi, C++ Builder, Power Builder (SY Base), Designer, Developer (Oracle), Visual Basic, Visual C++ және т.б.). Delphi файлының типтері. Delphi – де программаны байланыстыру және компиляциялау. Delphi компиляторының директивалары. Компиляция нәтижесі. Visual Studio аспаптары және оларды қолдану мүмкіндіктері.
- Жобалаудың визуальды ортасын ұйымдастыру. RAD - редактор, жөндегіш, формалар конструкторы. RAD өнімділігін мүмкіндіктеріне байланысты бағалау: визуальды өңдеу ортасы, программалау тілі, жобалаудың шаблондарының кітапханасы, компиляторлар, деректерді сақтау объектілері және олармен жұмыс істеу аспаптары, желілік программалау аспаптары және объектілері. Borland Delphi графикалық ортасының 10 маңызды функциялары. Меню құрамы. Бөлек меню пункттерін құру негізі және қызметі. Меню бөлімдерінің ағашы. Құру, ашу, жобаны сақтау немесе жобалар тобын және элементтерін смақтау. Меню пункттерімен және бөлімдерімен қолданылатын функциялар. Аспаптар бөлігі – өңдеу аспаптарының ағашы. Ішкі және сыртқы аспаптар. Image Editor, WinSight, BDE, BDE Administrator, BDSQL Links, SQL Builder, SQL Monitor, Data Pump, TeamSource, Code Insight, Repository және т.б. Қолданылатын басқыштар панелі және олардың қолданылуы. Жобады қолдану үшін объектілер панелі - құрамы, компоненттер жиыны және топтар жиыны, панелді жеке жобалаушыға құрастыру, компонентті пакеттеу. Визуальды редактор панелі – объектілер инспекторы. Компонент қасиеттеріне қатынауды ұйымдастыру. Уақиғалар беті және олардың компоненттер байланысының суретін қолдануы. Фокус түсінігі. Жобалау кезіндегі және деректерді өңдеу программаларын орындау кезіндегі объект активизациясы (фокустау). Формалар панелі – жоба контейнері. Бір және көпбеттік жобалар. Формаларды және панелдерді жобалау және орындау кезінде ауыстыру. Модальдық.

- Delphi файлдарының типтері. Формалар файлы (*.dfm), ресурстар файлы (*.res), пакеттер файлы (*.bpl), жобалар файлы (*.dpr), қосымшалар – модульдер файлы (unit's) кадрлар, компоненттер, жалпы қатынау файлдары (*.pas), жоба опциялары және жұмыс столын орнату (*.dof, *.dsk), резервтік көшірмелер – құрылымы, қызметі, құру және редакторлау аспаптары. Жоба менеджерлері және кітапханаларды өңдеу. Файл аттарының Delphi пакеттерімен байланысты кеңейтулері:
- **bpl** - runtime пакеті, .dll файл Delphi-ге адаптацияланған ОЖ. Негізгі аты .dprкнемесе .dprkwsourсе файлдарынан алынады.
- **Dcp** – пакет тақырыбынан және барлық .dcp пакет файлының конкатенациясынан тұратын, компилятор керек ететін символдарды қосатын бинарлы бейне. Қарапайым dcp file әрбір пакет үшін құрылады. Негізгі аты .dpr таратқышынан алынады. Пакеттерді қолданған кезде .dcp файлдар міндетті түрде болуы тиіс.

- **dcu and pas** - пакетке кіретін модулдер файлының бинарлы бейнесі. Әрбір unit үшін қажет файл.
- **dpk and dpkw** – таратқыш файлдар, программалаушымен құрылады. dpk және .dpkw пакеттері бірдей, бірақ dpkw кеңейтілуі кросс-платформалы қосымшалар үшін қолданылады, олар тек CLX кітапханасының компоненттерінен тұрады. Әдетте VCL және CLX кітапханаларының компоненттері қолданылады.
- Программаның орындалатын модулі келесілерден тұрады: тақырыптық бөлім, мұнда қосылатын кітапханалар аталады (uses); интерфейстік бөлім, мұнда қолданылатын деректер типі сипатталады, сонымен қатар класстар: орындалатын бөлім – имплементация, - мұнда ақпараттармен жаслатын әрекеттер сипатталады. C++ программалау кезінде тақырыптық (.h - header) файлдарда класстар сипатталады. Модулдерді байланыстыру #include командасымен орындалады.
- **Программның жөнделуі. Аспаптары. Жөндеу әдістемесі.**
- **Жөндеу процедурасы.** Жөндеу аспаптары. Бақылау нүктелері және өтулері. Жөндеу режимдері. Жөндеу кезіндегі қайталау әрекеттерінің минимизациясы. Жөндеуді басқару. Жөндеу құжаттары. Debuggers-ті қолдану – меню, мүмкіндіктер, командалар.
- Өңдеу кезіндегі қателерді іздеу – нәтижелерді түсінбеу, тоқтап қалулардың болуы, циклдан шықпау және программадан шығап кету кезінде көрінеді. design және run периодтық қателер. Жөндеу тәртібі – жүйелік хабарламаларды өңдеу, программадағы күдікті зоналарды табу, SHE фреймдердің қоршалуы және жүйелік хабарламаларды және өңдеуші хабарламаларын генерациялау, бақылау нүктелерін құру және күдікті аймақтарға тесттер құру үшін пайдаланылады.

- **Жөндеу аспаптары.** Декомпиляторлар, жөндегіштер және дизассемблерлер - құрамы, мүмкіндіктері, қолдану тәртібі. debuggers қолдану – менюі, мүмкіндіктері, командалары. Дизассемблирлеу, — процессордың екілік кодтарын түсінікті мнемоникалық инструкцияға келтіру. «IDA дизассемблері интенсивті даму өніміне жатады, — өңдеушілермен енгізілген үнемі даму арқасында және өзегрістердің арқасында көптеген версиялары пайда болған, солардың ішінде 3.84, 3,84b, 3,85, 4.0, ал кейбіреуілер әлі күнге дейін IDA 3.6 қолданғанды дұрыс көреді. Өкінішке орай, бір – біріне ұқсас версиялардың өзі ерекшеленеді және сәйкес келмейді. Дизассемблер пакетінің үш түрлі ұсыныстары бар — стандартты (IDA Pro Standard), прогрессивті (IDA Pro Advanced) және демонстрациондық (IDA Pro Demo). Мұндағы әрбір пакетке (демонстрациондықтан басқа) екі қосымша кіреді — бірі Windows-32 үшін графикалық (келешекте IDAG деп бергілейміз) және үш MS-DOS, OS/2 және Windows-32 үшін консольдық. Ал демонстрациондық пакетке тек бір ғана графикалық қана қосымша кіреді. Дизассемблердің екі түрі бар – автономды және интерактивті. Автономды пайдаланушыдан өзіне керекті шарттарды дизассемблерлеу процессінен бұрын беруді керек етеді және процесстің өзіне кірісуге рұқсат бермейді.Интерактивті дизаассемблерлеу кезінде программаның процессінің орындалуын "қолмен" басқаруға болады. Пайдаланушы кез – келген кезде процесстің орындалуына кірісе алады: адрестерді константалардан ерекшелеу немесе инструкцияның шегін көрсетуді анықтайды және т.б. Сәйкесінше, интерактивті дизассемблерлердің пайдаланушы интерфейсі жақсы дамыған, ал IDA–дың тіпті өз си-ұқсас скриптер тілі бар.

- **Бақылау нүктелері мен өзгерістері.** Программалардың құрылымдық және функциональдық декомпозициясы. Транзакциялар - қызметі, безендірілуі, қолданылуы. Есептеудің функционалдық бүтіндігін сақтау және деректердің бүтіндігі. Программаның орындалуын журнализациялау.
- **Жөндеу режимдері. Жөндеу кезіндегі қайталау жұмыстарын минимизациялау.** Автоматтандырылған және қолдан жөндеу. Трассировкалау және жұмыстың қадамдық орындалуы – микропроцессордағы және алгоритмдегі таратылуы. Айнымалылардың мәнін қарау және оларды оперативты түрде шығару. Қосылыстар (вставки) – бақылаушы және өзгертуші.
- **Жөндеуді басқару. Жөндеу құжаттары.** Коллективтік жұмыс құжаттары – кітапханалар, журналдар, оларды сақтау және басқару.