

ДӘРІС 11. DEBUGGER (ОРЫНДАУШЫ)-
ПРОГРАММАНЫ ОРЫНДАУҒА ЖІБЕРУ.
МҮМКІНДІКТЕР. ОРЫНДАУШЫНЫ ҚОЛДАУ.
ӨНДЕУДІҢ ВИЗУАЛДЫ ОРТАСЫН ЖІБЕРУШІЛЕР.
КЛАСТАР ДИАГРАММАСЫН ҚҰРУ ЖӘНЕ
ПАКЕТТЕРГЕ ТОПТАУ. ӘДІСТЕРІ,
ТЕХНОЛОГИЯЛАРЫ, АСПАПТАРЫ.

Мақсаты:

Кластар диаграммасын құру және пакеттерге топтау. Әдістері, технологияларын, аспаптарын үйрену.

- **Класс түсінігі.** Класс (Class) ортақ қасиеттері (атрибуттары), тәртібі (функциялары), семантикасы және басқа объектермен байланысы бар объектер тобын анықтайды. Класстың объектіні құруға арналған шаблон ретінде қарауға болады. Әрбір объект қандайда бір ғана кластың нұсқасы мысал ретінде келесідей сипаттамалары бар 'курсты ұсыну' класс анықтамасын қарастырайық :
- Атрибуттар – «сабақты өткізетін орын», «сабақты өткізетін уақыт», «курстың аталуы», «курс нөмірі», «ұсыныс нөмірі»;
- Функциялары – «сабақты өткізетін орынды анықтау», « сабақты өткізетін уақытты анықтау», «студентті тіркеу», «студентті тіркеуді тоқтату».
- «Курстық ұсыну» класының нұсқасының орнына 'алгебра 101, 1- бөлім' және 'алгебра 101 , 2- бөлім' объектілерінде алуға болатын еді, себебі олардың да әр қайсысының нақты атрибуттарының жиынтығы және ортақ функционалдық сипаттамалары бар.
- Дұрыс құрылған класс тек бір ғана абстракцияны бере алады. Мысалы, студент туралы мәлімет сақталған , сонымен қатар барлық оқу барлығында студент өткен курстар тізімі функциясы көрсетілген класты сәтті құрылған деп айта алмаймыз, өйткені ол екі әртүрлі операциялар тобын қамтиды. Сондықтан мұндай класты екіге – «студент» және «студент өткен курстар» бөлсе жақсы болар еді.

- Класстың атауы үшін пән облысына сәйкестендіріліп қабылданған терминдерді қолданған дұрыс. Класс аты ретінде жобаланатын түсінікті толығымен сипаттай алатын зат есіміңіз жекеше түрі қолданылады. Кейде қысқартылған атауларда қолданылып, класты құжаттандырғанда міндетті түрде мағынасын ашып көрсету керек. Егер аббревиатура бірдей мағыналы емес интерпретация жіберсе, онда сәйкесінше айтылудың толық түрін қолданамыз. Объектілер мен кластар арасындағы ерекшелікті табу қиын.
- UML-да класс зоналарға бөлінген тіктөртбұрыш түрінде кескінделеді. Жоғары зонада класс аты, ортасында оның құрылымы (атрибуттар тізімі) оның астындағысында тәртібі сипаттамаларын анықтайтын функциялар беріледі.
- Класты қалай құрады
- Browser терезесінің Logical View элементіне тышқан сілтеушіні қойып, контекстік менюді белсенді ету үшін оң батырмасын шертеміз.
- менюді New--- Class элементтерін таңдау; Browser терезесінде кескінделген ағаш (бұтақ) жаңа класқа сәйкес келетін New- Class элементімен толықтырылады.
- New-Class элементін таңдап және кластың қажет атауын енгізіп оны өзгертеміз.

- **Класстар диаграммасын алу тәртібі.**
Функциональдық және актерлармен функциялардың бірлесе қолданылуы. Ақпараттық элементтер және құрылымдар (АО) арасындағы қарым – қатынас және байланыс. Инкапсуляция, мұрагерлеу және АО полиморфизмі. Бірлесуі (уақыттық және жазықтықтық) және көрінуі (ішінде және сыртында). Декомпозициялық және құрылымдық. ҚВ, тізбекті, бірлескен (кооперативті) диаграммалары – орындалу тізбегі.
- **Класстар арасындағы қарым – қатынасты орнату.**
Агрегаттық қатынастар және мұрагер қатынастар. Қолдану қатынастары. Ассоциативті байланыстар. Бағыттық. Уақыттық және қалыпты байлансытар. Графикалық көрсетілуі.

- **RationalRose(RR)аспаптары және әдістемесі.** Логикалық жобалау автоматизациясының – артықшылықтары мен кемшіліктері. RRқұрылымдық құрамы. Компоненттік жүйелер және олардың керектігі мен программаларды өңдеудегі қолданылуы. Класстар шаблоныжәне олардың қолданылуы. Класстың тілдік көрсетілімі. RR класстар кітапханасы. CASE жүйелер.
- “CASE – құралдарына байланысты ақпараттар біздің ойымызша кәдімгі бір программалаумен тығыз байланысты зат сияқты. Америкада, конкуренцияның күшті болуына байланысты, CASE – құралдар программалық қамтаманы өңдеуші фирмаларды басып отыру үшін қолданылады. CASE – құралдардың дамуы объектілі – бағытталған ПҚ өңдеу технологияларына байланысты. Осы кезде объектілік моделдеу технологиялары да пайда бола бастады Booch, OMT, UML, олар өз алдына программалаумен байланыстыруға мүлдем келмейді. Бүгінде алдыңғы қатардағы CASE-жүйе болып Rational Rose корпорациясымен шығарылған Rational Software алынады. Rational Rose жүйесі Unified Modeling Language (UML) тілін қолдана отырып модулдер құру үшін пайдаланылады.Айталық, UML объектілі – бағытталған стандартты тілі болып қалыптасуы Rational Software – нің арқасында деп айтсақ та артық емес, ол UML-ді қолданатын программалық өнімдерді шығарып қана қоймай, сонымен қатар UML тілінің спецификациясын құратын және жаңартатын CORBA таратылған есептеулермен байланысты технологиялар Object Management Group (OMG) –пен де жұмыс жасайды, және Rational компаниясында үш, UML тілін және объектілі – бағытталған өңдеуді құрушылар жұмыс жасайды. Бұлар Гради Буч, Айвар Джекобсон және Джим Рамбаух.

- Rational Software Rational Rose 98 компаниясының CASE-жүйелері коммерциялық ПҚ құру үшін барлық жерлерде және әйгілі программалау тілдері Java, C++, Smalltalk, Ada, Visual Basic, Power Builder және Forte қолданылады. Сонымен қатар, Rose 98 пакеті CORBA и Data Definition Language (DDL) қосымшалары үшін, Interface Definition Language (IDL) тілдеріндегі сипаттамаларды генерациялауға, деректер базасына қатынау қосымшалары соның ішінде Oracle 8-ге қатынауға мүмкіндік береді. Айталық, сол немесе басқа программалау тілін қолдауы Rational Rose 98 пакетінің қай редакциясы екендігіне байланысты.
- Мысалы, қарапайым - Rose 98 Modeler Edition пакетіне қатты талаптар қоюға болмайды. Ал оның орнына Rose 98 Enterprise Edition барлық талаптарға сай деп айтсақ болады.
- Айталық, Rose жүйесі - визуальды моделдеу жүйесінде жақсы орын алады, оны қолдана отырып құрылып отырған қосымшаның құрылымын құруға, оның орындамалық текстін генерациялауға және параллельді өңделіп отырған жүйенің құжаттаумен жұмыс жасауға болады. Rational Rose көмегімен com модульдегі кері талдау базасы негізінде жаңа моделдер құруға немесе қолданбалы программа текстін және кітапханалар класстарын анықтауға болады.

RationalRose 98-дің артықшылықтары:

- Қосымшаның өңделу циклын қысқарту.
- Программист жұмысының өнімділігін арттыру.
- Бизнес және пайдаланушыларға байланысты тапсырыс берушілердің программа құрудағы сапалық бағасын жақсарту.
- Үлкен жобалар және жобалар тобын құруа алу мүмкіндіктері.
- Бұрын құрылған ПҚ қолдана алу және олардың құрылымы мен компоненттерін өзгерте алу мүмкіндіктері.
- UMLтілі әртүрлі бөлімдер мен өндірушілер арасындағы әмбебеп "көпір".
- Жүйенің кемшілігі, басқа да бөліктеп жинау жүйелері сияқты мұнда да көп керек емес бөліктер көп. Яғни RationalRose жобаның базалық құрамдас бөлігін жасайды.
- **Coad аспаптары және әдістемесі.** Программалық жүйе жобаларын өңдеу мысалы және әдістемесі. Тиімді объектілік моделдер құрудың практикалық құралдары - стратегиялары және бейнелері.
Стратегиялар: әрекеттер және компоненттер; жүйенің мінездемелік қасиеттерін және мақсаттарын анықтау; объектілерді таңдау; міндеттерін анықтау; сценарии көмегімен динамикасын өңдеу; жаңа стратегияларды және бейнелерді табу. **Бейнелер:** фундаментальды; транзакция; агрегаттар; жоспарлар; байланыстар.