

Технології розробки програмних засобів, які підтримують компонентно-орієнтований підхід

Співаковський О.В., Круглик В.С.

Херсонський державний університет

В даній статті мова йде про впровадження нових інформаційних технологій в процес вивчення Лінійної алгебри у Херсонському державному університеті. Розглядається архітектура системи та технології розробки.

Світовий процес переходу від індустріального до інформаційного суспільства, а також соціально-економічні зміни, що відбуваються в Україні, вимагають суттєвих змін у багатьох сферах діяльності держави. В першу чергу це стосується реформування освіти. Національною програмою "Освіта. Україна XXI" сторіччя передбачено забезпечення розвитку освіти на основі нових прогресивних концепцій, запровадження у навчально-виховний процес новітніх педагогічних технологій та науково-методичних досягнень, створення нової системи інформаційного забезпечення освіти, входження України у трансконтинентальну систему комп'ютерної інформації.

Для досягнення зазначених результатів необхідно швидкими темпами розвивати дистанційну освіту, запровадження якої в Україні передбачено Національною програмою інформатизації.

Особливо актуальним є застосування сучасних інформаційних технологій у тих сферах розумової діяльності, які є найбільш складні для сприйняття, коли складність навчання обумовлюється великою кількістю рутинної роботи. Велика кількість обчислень, яка супроводжує відшукування розв'язку тієї чи іншої задачі, не дає можливості учню засвоїти сутність досліджуваних процесів і явищ, і як наслідок – не формує необхідних знань та вмінь. У національній державній програмі "Освіта" (Україна XXI століття) зазначено, що освіта має забезпечувати всебічний розвиток людини як

цілісної особистості, її здібностей та обдарувань, збагачення на цій основі інтелектуального потенціалу народу, його духовності і культури, формування громадянина України, здатного до свідомого суспільного вибору [1].

Розвиток нових інформаційних технологій досяг того рівня, що можливим стала підтримка інформаційної сфери і поширення відповідних технічних засобів (комп'ютерних лабораторій, засобів телекомунікацій, оперативної поліграфії, систем інтерактивного відео, баз даних і т.д.), в навчальних закладах почався наступний етап цього процесу. Його основний зміст:

активне засвоєння і фрагментарне втілення засобів нових інформаційних технологій в традиційні навчальні дисципліни і на цій основі – масове засвоєння викладачами нових методів і організаційних форм навчальної діяльності;

побудова питання про радикальний перегляд змісту освіти, традиційних форм і методів навчально-виховної роботи;

розробка і засвоєння систем навчально-методичного забезпечення (“програмно-методичних комплексів”, “комп'ютерних курсів”), які містять програмні засоби для ЕОМ, різноманітні відео і аудіо матеріали, тексти для учнів і методичні матеріали для викладачів.

Багато які традиційні комп'ютерні курси математики, фізики та інших дисциплін базуються на ідеях програмованого навчання, хоча й використовують усі апаратні та програмні можливості сучасної обчислювальної техніки та нові методи подання знань. Найбільш розвиненою та досконалою як з методичної, так і з технічної точки зору при цьому виявляється лекційна частина курсу. Як правило, учбовий матеріал лекційної частини дисципліни супроводжується системами контрольних питань і тестових завдань. Так комп'ютерна система забезпечує контроль знань.

Але, як відомо усім педагогам, навчальний план дисципліни включає ще й практичну частину – цикли практичних занять і лабораторних робіт. Формування практичних вмінь та навичок досягається саме тут, і ця частина

навчального плану в багатьох дисциплінах (математика, фізика, інформатика, ...) є центральною. Іншими словами, студент знає математику, якщо він вміє вирішувати математичні задачі. Проблема адекватної комп'ютерної підтримки практичних занять менше розроблена і є актуальною [2].

Для дистанційного навчання важливе значення мають педагогічні програмні засоби. Для репродуктивного засвоєння навчального матеріалу, коли застосовуються друковані, аудіо- та відео- матеріали можливо обійтися і без педагогічних програмних засобів. В даному випадку дистанційне навчання в принципі виконує свою функцію при вивченні таких дисциплін, як історія, література, право, і т.п.

Але цього не досить при вивченні таких предметів, як математика, хімія, фізика, де важливу роль відіграють задачі. При вивченні таких дисциплін велику роль відіграє практична діяльність учнів.

В мережі Інтернет педагогічні програмні засоби з підтримкою практичної діяльності учнів зустрічаються дуже рідко, тому актуальною є проблема складання технології їх розробки та впровадження для дистанційного навчання.

Сучасна педагогічно-орієнтована інформаційна технологія повинна вирішувати задачі подання теоретичного матеріалу, формування практичних вмінь та навичок та контролю засвоєння теоретичних знань, практичних вмінь та навичок.

Технологія створення інтегрованих педагогічних програмних засобів для дистанційного навчання з підтримкою практичної діяльності учнів повинна задовольняти наступним вимогам:

- Ї Легкість реалізації;
- Ї Предметна незалежність;
- Ї Кроссплатформенність;
- Ї Легкість впровадження в існуючі програмні засоби;
- Ї Можливість роботи як в мережі Інтернет так і на комп'ютерах не з'єднаних з мережею.

Для реалізації даного підходу було створено програмну оболонку, яка має наступні особливості: легка інтеграція існуючих модулів, легка модернізація існуючих модулів, можливість розширення системи за рахунок нових модулів, легкість адміністрування. Розглянемо детальніше підходи, що застосовувалися при розробці.

Компонентний підхід до побудови систем дистанційної освіти

Для зменшення собівартості підготовки систем для дистанційної освіти, необхідно виділити загальні функції для всіх ППЗ та написати специфічні компоненти для кожної з них. При цьому важливо компоненти інтегрувати в оболонку, яка б підтримувала наступні функції: легка інтеграція існуючих модулів, легка модернізація існуючих модулів, можливість розширення системи за рахунок нових модулів, контроль прав доступу до модулів (читання та зміна).

Аналіз конкретної предметної області дозволяє виділити компоненти, які необхідні для правильної та повноцінної організації учбової діяльності. Надалі розглянемо більш детально виділені нами компоненти та їх реалізацію.

Компонент для реалізації входу до системи. За допомогою цього компоненту реалізується аутентифікація користувачів, та розподіл користувачів на групи.

Компонент безпеки. Використовується для авторизації користувачів та розподіл прав доступу в залежності від групи, до якої належить користувач.

Компонент адміністрування. Використовується для загального керування всім вузлом дистанційної освіти та організації взаємодії окремих компонент. Функціональність також дозволяє додавання, видалення, редагування інформації про користувачів, визначення прав користувачів на доступ до окремих компонентів.

Ці три модулі грають особливу роль, оскільки утворюють потужний каркас, який забезпечує легке впровадження та зручну взаємодію компонент, призначених для організації процесу навчання. Розглянемо тепер їх.

Компонент „Навчальний посібник”. Являє собою модуль для роботи з структурованим гіпертекстом. За допомогою нього студент отримує необхідну теоретичну допомогу.

Компонент „Задачник”. Є сховищем задач для конкретного предмету. Може бути структурно та функціонально об’єднаним з компонентом „Учбовий посібник”. Функціональні можливості дозволяють викладачам додавати та редагувати завдання, а студентам копіювати завдання до свого зошиту.

Компонент „Зошит”. Служить для зберігання задач студента та роботи з ними. Дозволяє отримати інформацію про те, чи вирішена задача, чи самостійно вирішена чи ні, оцінку за розв’язання. З „Зошита” студент також може завантажити задачу в середовище для розв’язування задач.

Компонент „Середовище для розв’язування задач”. Являє собою вузькоспеціалізований програмний модуль для розв’язку деякого класу задач конкретного предмету. Важливою особливістю середовища розв’язування слугує те, що вона повинна надавати інструментарій для рішення задач даної конкретної області та вміти перевіряти розв’язок на кожному кроці розв’язання, надаючи допомогу в разі необхідності. Зовнішній інтерфейс взаємодії з зошитом від зміни предмету змінюватися не повинен.

Компонент для проведення моніторингу. Дозволяє збирати данні та проводити статистичний аналіз для постійного контролю за ефективністю навчання.

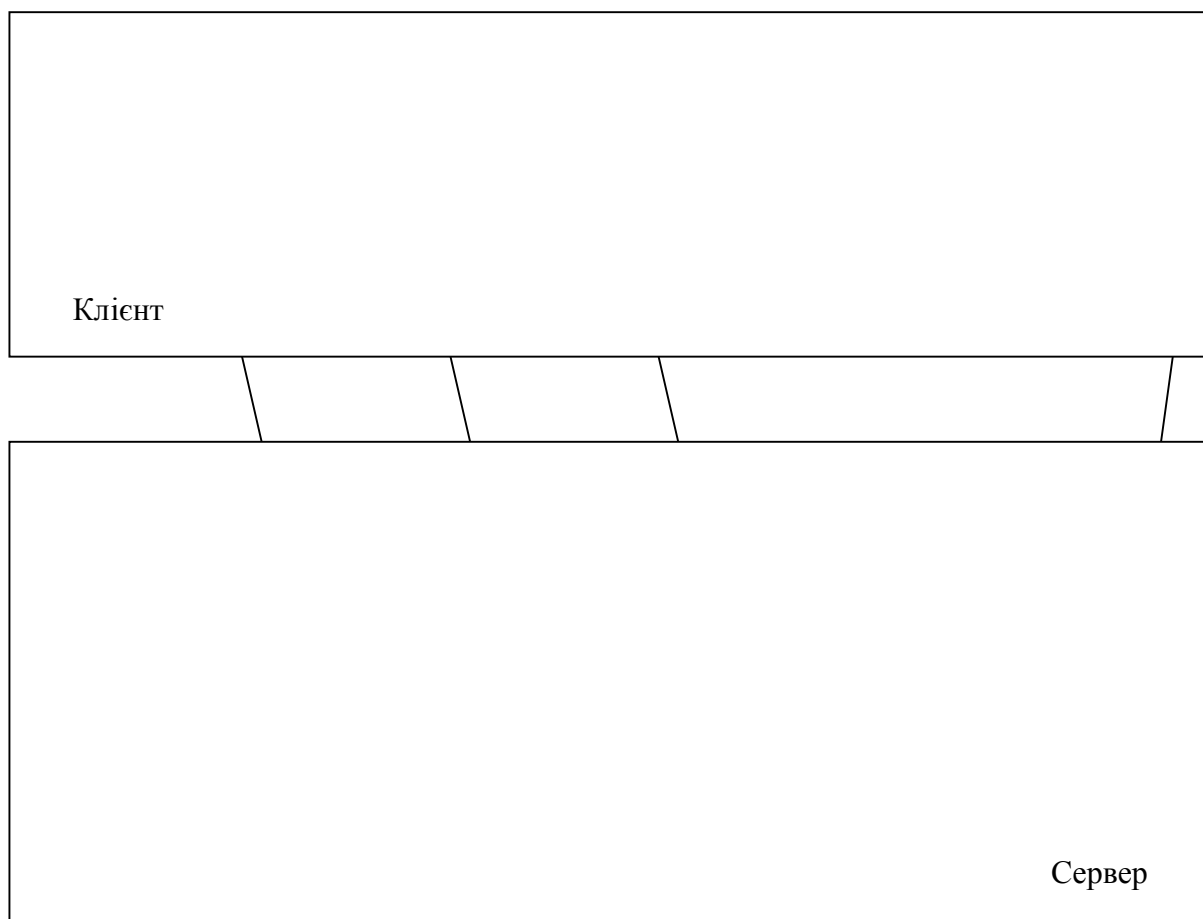
Компонент для групового обговорення проблем. Організується у вигляді форуму, де можна задати питання та отримати відповідь на нього.

Генератор завдань. Компонент призначено для генерації завдань в задачнику. Працювати з компонентом може лише викладач.

Для успішної реалізації наведених вище компонентів і розміщення їх в мережі Інтернет виберемо за основу архітектуру клієнт-сервер. В даному випадку програмою клієнтом є Web – браузер. Постає питання, які з компонентів потрібно розмістити на сервері, а які на клієнті, а також де саме

розмістити дані, з якими працює користувач. Ці питання були вирішені наступним чином. Всі дані, з якими працює користувач, розміщені і зберігаються на сервері. Клієнт лише має засоби для контролю над даними, інструментарій для управління даними. До клієнтської частини було віднесено такі компоненти: середовище для розв'язування задач та генератор завдань. Такий вибір викликаний з вимогою до швидкодії. Якщо ці компоненти розміщені на клієнті, то всі операції над даними (формування та вирішення задачі) відбуваються без передачі даних на сервер, що дуже важливо для підвищення швидкодії при повільному Інтернеті.

Схема взаємодії клієнт-сервер, що використовується



Вибір технологій для розробки курсу дистанційної освіти

Дистанційні курси для мережі Інтернет можуть розроблятися використовуючи веб-броузер в якості клієнта. Важливим є питання про вибір найкращої існуючої технології, яка б забезпечувала необхідну

інтерактивність та можливості об'єктно-орієнтованого програмування, можливість повторного використання коду. Крім того, розробка та подальша підтримка не повинна бути надто складною.

Для розробки курсу було вибрано такі інструментальні засоби: Microsoft Visual Studio 2003, Delphi 6.0; графічні редактори: CorelDraw 11, Adobe PhotoShop 7.0; підготовка текстів – Microsoft Word XP.

Операційна система – Microsoft Windows XP Professional.

Web-сервер – Microsoft Internet Information Server.

Серед обраних технологій можна назвати ASP.NET, XML, XSL, ADO.NET, COM+, ActiveX.

Розглянемо детальніше технологію ASP.NET, її переваги над іншими існуючими технологіями програмування для Інтернет.

По-перше ASP.NET є системою, яка повністю підтримує концепції об'єктно-орієнтованого програмування.

По-друге ASP.NET є системою, що підтримує модель програмування, засновану на подіях.

По-третє в ASP.NET є вбудована підтримка XML формату даних.

ADO.NET в свою чергу забезпечує добру роботу з базами даних.

В якості мови програмування було вибрано C#. Саме C# є рідною мовою середовища .NET Common Language Runtime. Разом з .NET Common Language Runtime ця мова може використовуватися для написання об'єктно - орієнтованих програм з залученням всього досвіду програмування на C, C++ та COM. Тобто мова C# призначена для побудови надійних компонентів, які використовуються в практичних ситуаціях.

Середовище .NET Common Language Runtime побудована на компонентній базі. C# є „компонентно – орієнтованою” мовою, в тому смислі, що всі об'єкти оформлюються у вигляді компонентів і все, що відбувається так чи інакше зв'язано з компонентами.

Такі концепції компонентів, як властивості, методи та події, мають першочергове значення для мови та базового середовища виконання. З

компонентами може бути зв'язана декларативна інформація (атрибути), яка передає відомості про компоненти іншим частинам системи на стадії конструювання чи виконання. Існує можливість впровадження документації в компоненти та її експортування в XML.

“Garbage Collector” знімає з програміста важку роботу управління пам'яттю.

Програмування на C# відрізняється простотою, надійністю та логічністю. Механізм обробки помилок в ньому є дуже потужним, виключні ситуації оброблюються практично на будь-якому рівні. C# є безпечним по відношенню до типів даних. Він захищає користувача від непроініціалізованих змінних, небезпечних приведень типів та інших розповсюджених помилок програмістів.

Програмування давно покинуло область „чистої науки”. Програми рідко будуються на пустому місці, вони повинні відповідати деяким вимогам до швидкодії, інтегруватися з існуючим програмним кодом, а їх написання повинно оправдуватися з позицій затрат часу та грошей. Навіть добре спроектоване середовище не приносить особливої користі, якщо воно не має достатніх можливостей для практичного використання.

C# має всі переваги уніфікованого середовища, але при цьому надає доступ до інших можливостей, якщо в цьому є необхідність.

C# зберігає зусилля, затрачені на розробку готового програмного коду. Існуючі об'єкти COM можуть використовуватися так, наче вони є об'єктами .NET . Із програмного коду C# можна звернутися до коду C, який знаходиться в файлах DLL.

При необхідності C# дозволяє використовувати низькорівневі засоби. Низькорівневий доступ реалізується в режимі `unsafe`, який дозволяє використовувати вказівки в ситуаціях, коли швидкодія є критичним фактором або вказівки необхідні для взаємодії з існуючими DLL .

На кінець, C# широко використовують можливості .NET Common Language Runtime, котра забезпечує потужну бібліотечну підтримку для

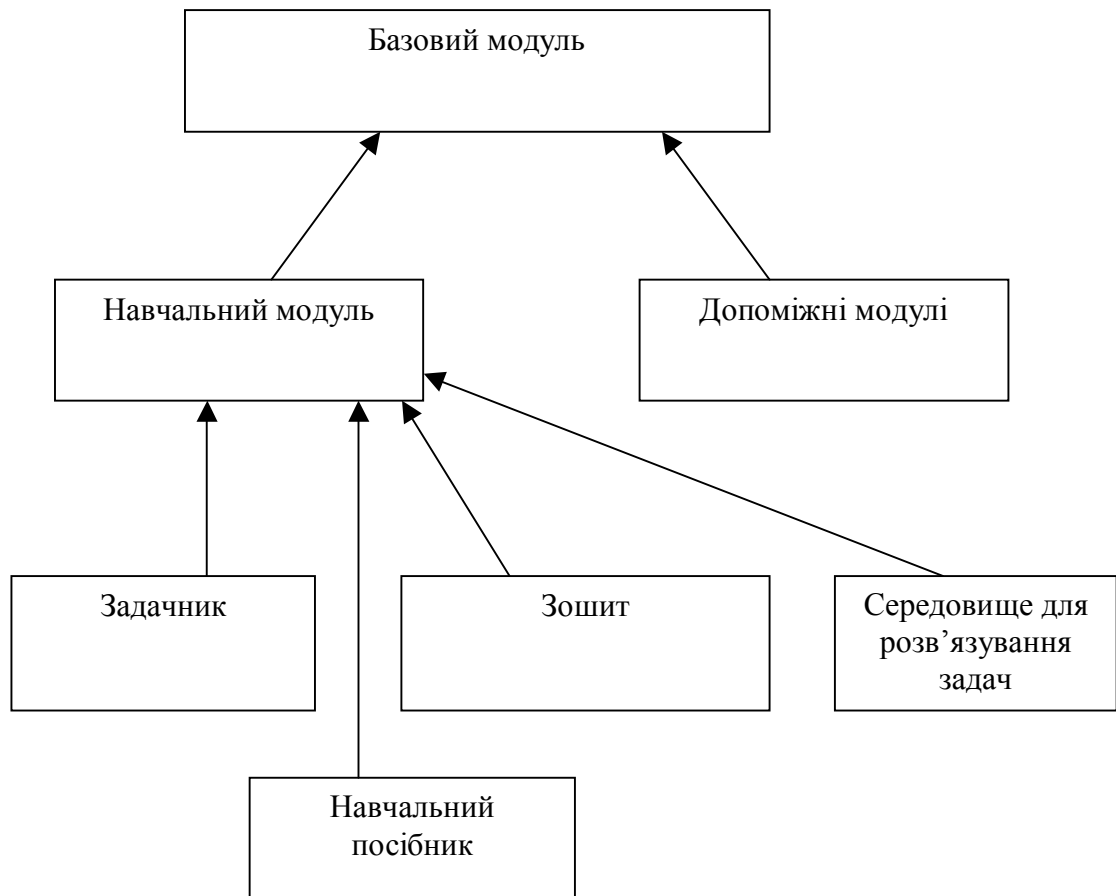
вирішення загальних та спеціалізованих задач програмування. .NET Common Language Runtime, Frameworks та мови програмування об'єднуються в середовищі Visual Studio, котра надає всі необхідні засоби в розпорядження програміста .NET .

Представлення даних в форматі XML дає необхідну універсальність та кроссплатформеність при зберіганні та обробці даних. XML є текстовим форматом описання даних. Він приніс нові методи передачі даних в Web додатках, дозволяючи передавати та використовувати практично любі типи даних де завгодно. Вся цінність XML витікає всього з двох простих особливостей. Перша особливість – XML можна розширювати. Це значить, що його можна легко доповнити, додавши власні структури та дескриптори користувача. По-друге, його формат є текстовим, що дозволяє створювати та редагувати його в будь-якому текстовому редакторі, наприклад в Блокноті.

Використання перетворень XSL дає можливість різного представлення даних на смак користувача, використовуючи при цьому як файл даних використовується XSL.

Основою будь-якої корпоративної системи є сховища даних. Тому правильний вибір типу сховища гарантує високу швидкість та стабільність роботи системи. В нашому випадку як корпоративне сховище даних було обрано Microsoft SQL Server 2000. Ця СУБД відповідає усім вимогам щодо швидкості, безпеки, легкості адміністрування. Використання збережених процедур сприяє швидкості виконання запитів та зменшенню кількості помилок при роботі з системою взагалі.

Ієрархія модулів для використання в системах дистанційного навчання



Модулі реалізовані у вигляді компонентів користувача системи ASP.NET.

Компоненти користувача системи ASP.NET – це сукупність елементів HTML, які об'єднані загальною функціональністю. Вони служать для візуального представлення інформації користувачам. Формування даних проходить в компонентах бізнес логіки, які можуть безпосередньо зв'язуватися з базою даних. Таким чином можна побудувати наступний ланцюг передачі даних:



Таким чином використовується 3-х рівнева архітектура клієнт-сервер.

Компонент „Середовище для розв'язування задач”

„Середовище для розв’язування задач” (надалі СРЗ) являє собою уніфіковане середовище, розроблене під певну предметну область, яке надає необхідний інструментарій для вирішення задач, проведення віртуальних експериментів в даній предметній області.

До СРЗ ставляться наступні вимоги:

Програмні вимоги:

- Ї Універсальність використання;
- Ї Переносність;
- Ї Повторне використання коду;
- Ї Стандартизація інтерфейсу взаємодії;

Вимоги до інтерфейсу:

- Ї Інтерфейс користувача повинен бути інтуїтивно зрозумілим;
- Ї Інтерфейс користувача повинен бути максимально наближеним до використання в даній предметній області;
- Ї Повинен бути набір команд користувача, за допомогою яких можна робити всі допустимі перетворення над об’єктами даної предметної області.

Розглянемо більш детально програмні вимоги. Універсальність використання та переносність вимагає, щоб розроблений продукт легко було впроваджувати в існуючі програмні середовища, як в локальні версії, так і в версії, які орієнтовані на мережу Інтернет. Повторне використання коду передбачає впровадження продукту без повторної перекомпіляції та без внесення будь яких змін в коди.

Стандартизація інтерфейсу взаємодії дозволяє вирішити наступні проблеми:

- Ї Незалежність коду від розробника (тобто компоненти може розробляти будь-хто, наслідувавши інтерфейс від існуючого);
- Ї Можливість розробляти програмні засоби під ще не існуюче СРЗ;
- Ї Легкість локалізації та усунення помилок.

Щоб виконати наведені вимоги було вибрано в якості СРЗ використовувати елементи управління ActiveX з наперед визначеним інтерфейсом.

Розглянемо детальніше найважливіші методи, події та властивості, які надає даний інтерфейс.

Методи:

Load – метод призначений для завантаження задачі в СРЗ;

Save – метод призначений для збереження задачі;

Властивості:

Task – властивість призначена для завантаження та збереження задач;

Події:

OnSave – подія виникає при збереженні задачі.

В якості формату взаємодії вибрано XML. Тобто кожна задача представляє собою опис своєї структури мовою розмітки даних XML.

Для того, щоб використати розроблений елемент управління ActiveX СРЗ в будь-якій програмі необхідно його створити, потім викликати метод **Load**, куди в якості параметра необхідно передати XML – текст, який описує задачу. З цього XML тексту середовище повинне повністю відновити ту ситуацію, коли дана задача була збережена.

Коли користувач зберігає задачу, виникає подія **OnSave**. Формат - **OnSave(xmlstr As String)**. В **xmlstr** передається вже сформований XML – текст задачі.

Збереження задачі можна зробити також і з контейнера. Для цього необхідно викликати метод **Save**.

Завантаження та збереження задач можна виконувати за допомогою властивості **Task**. При запису у властивість відбувається завантаження задачі, при читанні – збереження задачі.

При завантаженні задачі до середовища для розв'язування задач необхідно перетворити вхідний формат з мови XML у внутрішній формат

даних, який в подальшому буде використовуватися при обробці даних, тобто перетворюватися у процесі вирішення задач.

Реалізація середовища розв'язання задач для курсу „Лінійна Алгебра”

Середовище для розв'язування задач курсу „Лінійна алгебра” є уніфікованим середовищем для розв'язування задач з лінійної алгебри. Прототипом даного середовища є середовище, розроблене під операційну систему MS-DOS Співаковським О.В., яке вже протягом тривалого часу використовується при проведенні практичних занять на фізико-математичному факультеті Херсонського державного університету.

Середовище повністю підтримує розв'язування наступних задач над полем раціональних чисел:

- Ї Розв'язати систему лінійних рівнянь;
- Ї Знайти визначник матриці;
- Ї Побудувати обернену матрицю;
- Ї Знайти характеристичний многочлен;
- Ї Знайти власні вектори;
- Ї Знайти ранг матриці;
- Ї Побудувати жорданову форму;
- Ї Ортогоналізувати лінійну оболонку.

Для вирішення наступних задач передбачено наступні команди:

- Ї Скласти дві матриці;
- Ї Помножити матрицю на число;
- Ї Перемножити дві матриці;
- Ї Транспонувати матрицю;
- Ї Одночасно перетворювати матриці.
- Ї Операції над матрицею:
- Ї Відмітити рядок;
- Ї Відмітити стовпчик;
- Ї Скласти відмічені рядки;

- Ї Переставити відмічені рядки;
- Ї Переставити відмічені стовпчики;
- Ї Видалити рядок;
- Ї Видалити стовпчик;
- Ї Вставити нульовий рядок;
- Ї Вставити нульовий стовпчик.

В „Середовищі для розв’язування задач” існує можливість інтерактивної допомоги – режим роботи „Експерт”. Цей режим вмикається студентом в ситуації, коли він не може продовжити розв’язання задачі. В цьому випадку програма оцінює існуючу ситуацію та виконує наступний крок рішення. Коли студент повністю вирішує задачу, експерт перевіряє правильність розв’язання задачі та повідомляє – „Задача не розв’язана”, „Задача розв’язана несамотійно”, „Задача розв’язана самотійно і неправильно”, „Задача розв’язана самотійно і правильно”. Інформація про хід та результати розв’язування задачі надалі зберігаються в зошиті.

В даному середовищі також реалізовано повністю „мишиний інтерфейс”, тобто працювати з середовищем можна використовуючи лише один маніпулятор „миша”. Наприклад, додавання рядків відбувається перетягуванням рядка на рядок, введення множників за допомогою екранної клавіатури.

Застосування „мишиного інтерфейсу” та режиму роботи „Експерт” роблять роботу з середовищем надзвичайно легкою та інтуїтивно зрозумілою.

Середовище повністю підтримує принципи компонентно-орієнтованого навчання.

Впровадження середовища в Інтернет-оболонку дозволяє використовувати його як в мережі Інтернет так і в локальній мережі університету.

Розроблена модель курсу «Лінійної алгебри» забезпечує:

- Ї виконання державного освітнього стандарту навчання «Лінійної алгебри»;
- Ї поглиблення і розширення теоретичної бази курсу, в першу чергу за рахунок скорочення часу на використання рутинних операцій;
- Ї використання систем віддаленого доступу до структурованої навчальної інформації для студентів і викладачів, як у синхронному, так і в асинхронному режимах;
- Ї можливість виділення навчальних одиниць, які можуть використовуватися як компоненти розв'язування задач більш високого рівня складності;
- Ї використання комп'ютерно-орієнтованих програм навчального і професійного призначення під час вивчення курсів вищої математики; підвищення практичної значущості результатів навчання;
- Ї створення умов для максимально повного розкриття здібностей студентів, формування необхідного рівня мотивації навчальної діяльності.

Література

1. О.В. Співаковський . Підготовка вчителя математики до використання комп'ютера у навчальному процесі. //Комп'ютер у школі та сім'ї.-1999.-№2(6), с. 9-12.
2. Співаковський О.В., Львов М.С., Кравцов Г.М., Крекнін В.А., Гуржій Т.А.Зайцева Т.В.Кушнір Н.А., Кот С.М. Педагогічні технології та педагогічно–орієнтовані програмні системи: предметно–орієнтований підхід. // Комп'ютер у школі та сім'ї. – 2002.– №2 (20). – С. 17–21.
3. Співаковський О.В., В.А. Крекнін. Лінійна алгебра: Навчальний посібник. – Херсон: Айлант, 1997.– 148 с.
4. Співаковський О.В., Крекнін В.А., Черниш К.В. Збірник задач і вправ з лінійної алгебри: Навчальний посібник. – Херсон: Айлант, 2000–206 с.