

ally conjugate rotors developed programs written object-oriented language Borland Pascal with Objects 7.0 for calculations and language AutoLisp to form in AutoCAD the spatial images simulated rotors.

Key words: geometric modelling, visualization, rotors are mutually conjugate, Roots-type blower.

Ирина УСТЕНКО

г. Николаев

МОДЕЛИРОВАНИЕ ВЗАИМОСOPЯЖЕННЫХ РОТОРОВ ВОЗДУХОДУВОК ТИПА РУТС

В работе рассмотрены вопросы геометрического моделирования и визуализации профилей головок и впадин трехлопастных роторов воздуходувок типа Рутс. На базе предложенного метода геометрического моделирования взаимосопряженных роторов разработаны программы, написанные объектно-ориентированным языком Borland Pascal with Objects 7.0 для проведения расчетов и языком AutoLisp для формирования в среде проектирования AutoCAD пространственного изображения смоделированных роторов.

Ключевые слова: моделирование геометрическое, визуализация, роторы взаимосопряженные, воздуходувка типа Рутс.

Стаття надійшла до редколегії 06.03.2016

УДК 004.413

Сергій УСТЕНКО, Владислав ПРАДЕДОВ

м. Миколаїв

ustenko.s.a@gmail.com, sandapter@gmail.com

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ОБЛІКУ ТА АНАЛІЗУ ПОТОЧНОЇ УСПІШНОСТІ СТУДЕНТІВ

Робота присвячена розробці програмного забезпечення у вигляді сервіс-орієнтованого програмного додатку «Електронний журнал» для автоматизації обліку та аналізу поточної успішності студентів. Об'єктами обліку будуть відвідування занять, оцінки з дисциплін за певними видами контролю (з відміткою дати отримання) та показники готовності студентів до здачі сесії. До електронного журналу матиме доступ тільки викладач і його можна буде заповнити тільки під час проведення занять, що спонукає викладача відмічати присутніх і виставляти оцінки своєчасно, оскільки в іншому випадку заняття буде вважатись пропущеним і йому не буде зараховане навчальне навантаження.

Ключові слова: розробка програмного забезпечення, автоматизація обліку та аналізу, поточна успішність студентів, сервіс-орієнтований додаток, служба, клієнт.

Одним з основних завдань навчальної діяльності вищих навчальних закладів є поточний контроль успішності студентів, періодичність якого може змінюватися в залежності від розв'язуваних управлінських задач. Аналіз результатів поточного контролю дозволяє керівництву навчального закладу здійснювати оперативне управління процесом навчання студентів, процесом протікання сесійного контролю, а також підвищити мотивацію студентів і, як наслідок, якість освіти. Для розв'язання таких задач доцільне введення електронного обліку успішності студентів за допомогою електронного журналу.

У більшості випадків проведення аналізу результатів поточного контролю успішності вимагає значних тимчасових витрат, пов'язаних з необхідністю ретельної організації процесів збору і обробки великого обсягу оперативної інформації. В роботі [6] розглянуто процес розробки онтології підсистеми оперативного контролю поточної успішності студентів.

При аналізі існуючих рішень були розглянуті наступні поширені реалізації електронного журналу:

- програмний модуль «ПС-Журнал успішності-Web» (www.politek-soft.kiev.ua/index.php?do=newdevelopments&product=ps-gradebook-web);
- web-сервіс «Електронний журнал» автоматизованої системи управління навчальним закладом (mkr.org.ua/portalinfos/index/1/21);
- Free Dean's Office (Електронний деканат) – модуль для середовища дистанційного навчання Moodle (www.deansoffice.ru).

Існують і інші реалізації електронних журналів, які є закритими і вимагають авторизації, наприклад електронний журнал комп'ютерної академії «Шаг» (new.logbook.itstep.org/login). Описи згаданих систем, як і інші джерела, не містять інформації про моделювання предметної області електронного журналу.

У даній роботі пропонується розробити сервіс-орієнтований програмний продукт, який дозволить автоматизувати задачу обліку поточної успішності студентів, оперативного контролю поточної успішності студентів і дозволить викладачеві працювати з електронним журналом в режимі реального часу.

Кожен семестр складається розклад і вноситься в базу даних електронного журналу. Крім того, в базі даних повинна бути актуальною інформація про студентів.

Користувачами програмного продукту «Електронний журнал» будуть викладачі, які є відповідальними за здійснення контролю знань студентів. Дані користувачі матимуть доступ до відвідуваності та зможуть редагувати й виставляти оцінки студентам.

Загальна схема розроблювального програмного продукту наведена на рис. 1. Він буде складатись зі служби, що буде реалізована у вигляді консольного додатку, та клієнта – WinForms-додаток. Служба буде мати дві кінцеві точки (endpoint) для підключення клієнта (MEX та epLogin). Між службою і клієнтом буде встановлено дуплексний режим підключення.

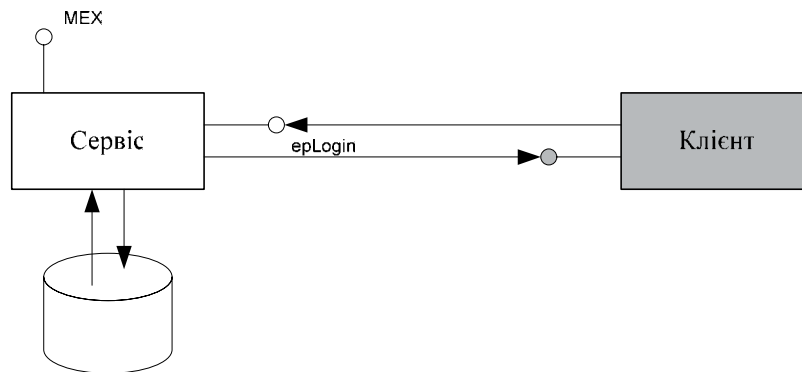


Рис. 1. Архітектура програмного продукту

Крім того, служба буде підключатись до СУБД SQL Server з використанням технології ADO.NET. Клієнт (викладач) матиме наступні можливості:

- підключатись до служби, якщо у нього у розкладі є заняття;
- завершувати роботу зі службою;
- отримувати інформацію про студента;
- відмічати присутність студента на занятті;
- виставляти оцінки студентів.

Служба, завдяки дуплексному режиму підключення, буде періодично поновлювати інформацію у клієнта, зокрема сповіщати його, що почалося або закінчилося заняття.

Таким чином, в результаті аналізу вимог до програмного продукту виявлені наступні сутності:

- розклад занять;
- заняття;

- дисципліна;
- викладач;
- студенти;
- оцінки.

Між даними сутностями існують зв'язки, які можна розділити на дві групи: «один до багатьох» або «багато до одного» (викладач і розклад занять, дисципліна і розклад занять, розклад занять і заняття) та «багато до багатьох» (заняття і студенти, студенти і оцінки). Названі сутності та зв'язки між ними показані на рис. 2.

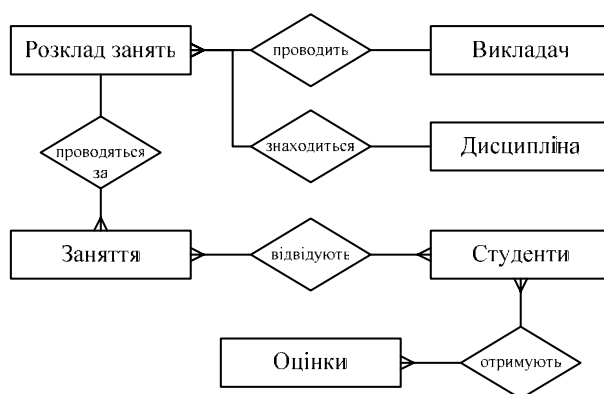


Рис. 2. Концептуальна модель даних

Виходячи із правил перетворення концептуальної моделі в логічну, включаючи перетворення зв'язків багато до багатьох, остання буде мати такий вигляд, як наведено на рис. 3.

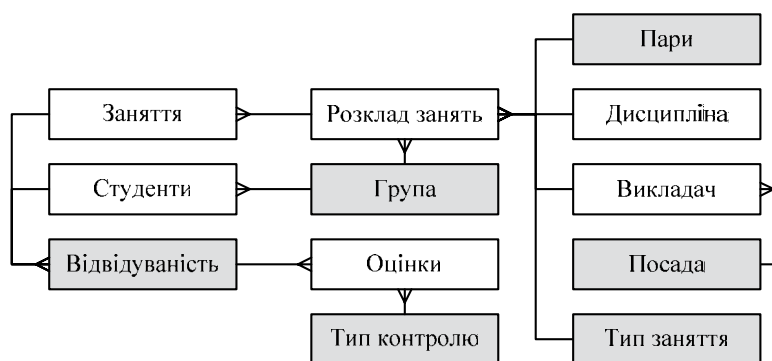


Рис. 3. Логічна модель даних

Якщо порівняти концептуальну модель даних та логічну, то можна побачити, що з'явилося 6 таблиць, що не відповідають сутностям концептуальної моделі, з яких 5 це довідники (пари, група, посада, тип заняття, тип контролю), а шоста – зв'язувальна таблиця «Відвідуваність», за допомогою якої були замінені два зв'язки «багато до багатьох».

Оскільки вище були виконані роботи зі створення логічної моделі бази даних, то можна очікувати, що отримані відношення будуть знаходитись у 2-й і 3-й нормальній формі [8].

Інтерфейс користувача програмного продукту має велике значення. Він повинен бути зручним в роботі користувачу програмного додатку, повинен відображати таку кількість інформації, яку користувач без проблем може сприйняти [1, 5, 10].

При проектуванні інтерфейсу необхідно враховувати ряд особливостей, з якими може зіткнутися користувач програмного забезпечення, такі як розподільча здатність монітору, розмір шрифту тощо. Інтерфейс не повинен містити дуже велику кількість графічних зображень, оскільки це сильно впливає на ресурси додатку.

Інтерфейс користувача для служби буде дуже простий, оскільки служба буде реалізована у вигляді консольного додатку. В процесі роботи служби на консоль буде виводитись інформація про основні події, що відбуватимуться в службі:

- спроба клієнта авторизуватися та її результат;
- запит інформації про студента;
- відмічання присутності студента на занятті;
- виставляння оцінки студентові;
- завершення роботи клієнта.

Клієнт буде мати графічний інтерфейс користувача. Схематично він буде мати вигляд, як на рис. 4. Схема містить 6 основних елементів:

- 1) панель авторизації;
- 2) панель інформації про заняття;

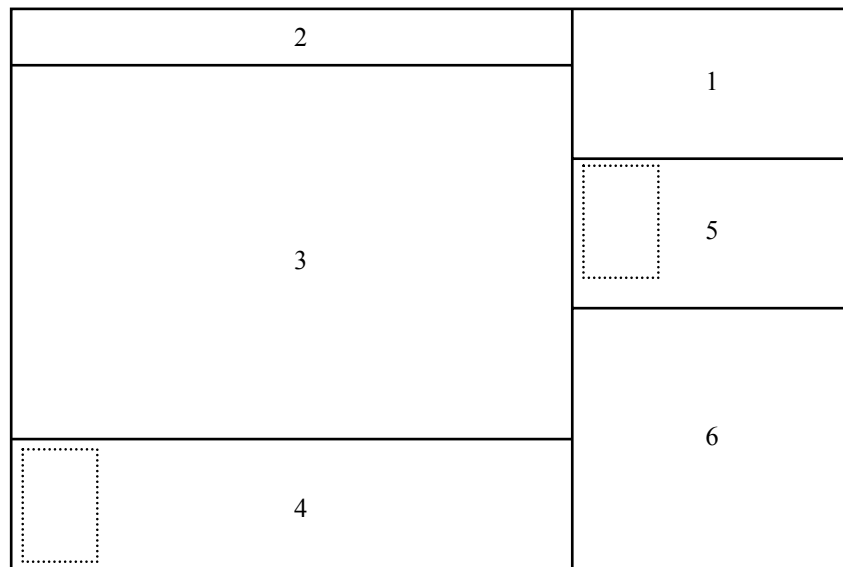


Рис. 4. Схема інтерфейсу користувача клієнта

- 3) список студентів з відміткою при присутність на занятті;
- 4) інформація про викладача;
- 5) інформація про студента вибраного на панелі 3;
- 6) інформація про відмітки отримані студентом на даному занятті.

Інтерфейс користувача буде мати резиновий характер, тобто при зміні розмірів вікна будуть зберігатися основні співвідношення, а саме:

- 1 і 5 елемент будуть залишатися незмінними;
- у 6 елементу незмінною буде тільки ширина;
- у 2 та 4 незмінною буде висота;
- 3 елемент буде займати весь простір, що залишиться.

Для розробки служби потрібно було спочатку розробити інтерфейс взаємодії з клієнтом. На основі вимог до «Електронного журналу», сформульованих вище, було розроблені такі контракти (з урахуванням того, що потрібно забезпечити дуплексний режим):

```
[ServiceContract(CallbackContract=typeof(IJournalCallback))]
interface IJournal
{
    [OperationContract]
    bool IsLogin(string login, string password);
    [OperationContract(IsOneWay = true)]
    void Exit();
    [OperationContract(IsOneWay = true)]
    void SetPresent(int idstudent, bool present);
}
```

```

[OperationContract]
StudentInfo GetStudent(int idstudent);
[OperationContract(IsOneWay = true)]
void SetMark(int idstudent, int idforwhat, int mark);
}

interface IJournalCallback
{
[OperationContract(IsOneWay=true)]
void Schedule(byte[] sched, PairInfo info);
}

```

Як було вже сказано раніше, служба має дві кінцеві точки для підключення клієнта: MEX – стандартний контракт IMetadataExchange для автоматичного налагодження клієнта та epLogin – створений IJournal.

Розроблено клас служби для роботи з клієнтом Journal, що наслідує інтерфейс IJournal та реалізовує його методи, а також створено клас JournalSender, за допомогою якого реалізується дуплексний режим передачі даних, а саме викликається функція оберненого виклику реалізована на клієнті.

У клієнта розроблено клас Server, в якому реалізовано функцію оберненого виклику Schedule, а також обробники події для графічного інтерфейсу користувача.

Всі інші класи, завдяки MEX-точці, автоматично додані до клієнтського проекту програмного продукту. Виклик потрібних методів інтерфейсу IJournal здійснюється за допомогою посередника класу JournalClient, створеного автоматично.

На основі вище наведеного розроблено програмне забезпечення сервіс-орієнтованого додатку «Електронний журнал» об'єктно-орієнтованою мовою програмування C# і складається з двох частин: служби та клієнта.

Служба створена з використанням платформи WCF, завантажується на одному з комп'ютерів локальної мережі факультету або кафедри і має вигляд консольного додатку. На рис. 5 наведено знімок вікна консолі служби при роботі двох викладачів з електронним журналом.

```

C:\Windows\system32\cmd.exe
Служба журналу запущена (для завершення натисніть Enter)...
Підключення до MS SQL Server...
Спроба авторизації...
Авторизовано xela...
Журнал для 6 запущено...
Видано інформацію про студента 70 (6)...
У журналі відмічено студента 70 (6)...
Видано інформацію про студента 70 (6)...
Видано інформацію про студента 70 (6)...
Видано інформацію про студента 73 (6)...
У журналі відмічено студента 73 (6)...
Видано інформацію про студента 73 (6)...
Видано інформацію про студента 73 (6)...
Видано інформацію про студента 70 (6)...
У журналі відмічено студента 70 (6)...
Видано інформацію про студента 70 (6)...
Видано інформацію про студента 70 (6)...
У журналі відмічено студента 70 (6)...
Видано інформацію про студента 70 (6)...
Видано інформацію про студента 70 (6)...
Видано інформацію про студента 71 (6)...
Видано інформацію про студента 72 (6)...
Видано інформацію про студента 73 (6)...
Підключення до MS SQL Server...
Спроба авторизації...
Авторизовано valery...
Журнал для 1 запущено...
Видано інформацію про студента 87 (1)...
Видано інформацію про студента 81 (1)...
У журналі відмічено студента 81 (1)...
Видано інформацію про студента 61 (1)...
Видано інформацію про студента 68 (6)...
Видано інформацію про студента 62 (6)...

```

Рис. 5. Робота служби

При розробці клієнтського додатку потрібно спочатку додати посилання на службу для цього потрібно вибрати пункт меню Project => Add Service References..., а вікні, що з'явиться, ввести адресу МЕХ-точки.

Після цього в проекті з'являться всі необхідні для роботи класи, в тому числі і посередник JournalClient.

JournalClient host = new JournalClient(Server.site);

Далі потрібно реалізувати функцію оберненого виклику Schedule та реалізувати роботу з інтерфейсом користувача.

На рис. 6 наведено знімок вікна клієнта при роботі з електронним журналом.

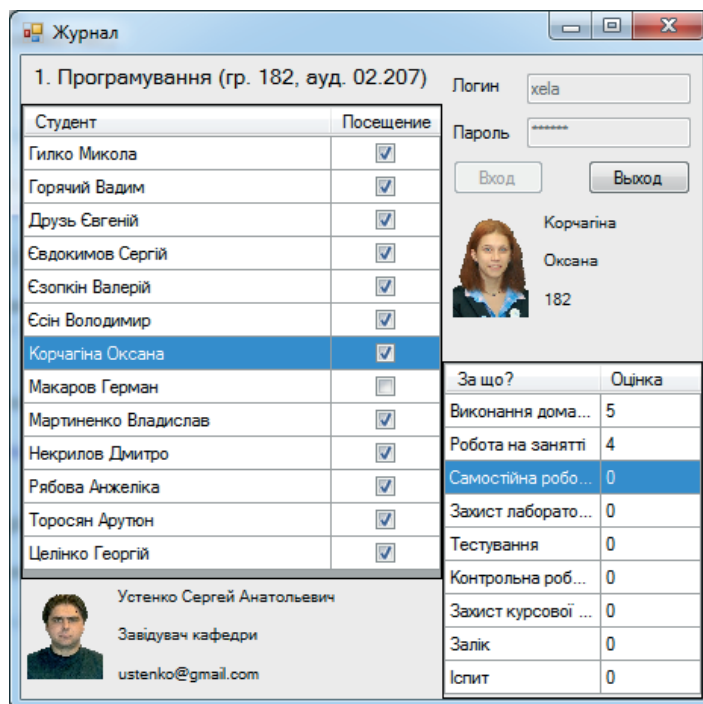


Рис. 6. Робота клієнта

Розроблене програмне забезпечення у вигляді сервіс-орієнтованого додатку Електронний журнал призначене для автоматизації обліку відвідування занять та успішності студентів, що дозволяє викладачеві під час проведення занять відмітити присутніх і виставити їм оцінки. Надалі розроблений програмний продукт планується інтегрувати до запланованої до реалізації системи управління університетом, в якій буде використовуватись спільна база даних.

Список використаних джерел

1. Акчурин Э. А. Человеко-машинное взаимодействие / Э. А. Акчурин. — М. : СОЛОН-ПРЕСС, 2008. — 96 с.
2. Бокс Д. Основы платформы .NET, Том 1. Общезыковая исполняющая среда / Д. Бокс, К. Селлз. — М. : Издательский дом Вильямс, 2003. — 288 с.
3. Бошешин Б. Основы ADO.NET / Б. Бошешин. — М. : Издательский дом Вильямс, 2003. — 448 с.
4. Вилдермьюс Ш. Практическое использование ADO.NET. Доступ к данным в Internet / Ш. Вилдермьюс. — М. : Издательский дом Вильямс, 2003. — 288 с.
5. Гульяев А. К. Проектирование пользовательского интерфейса / А. К. Гульяев, В. А. Машин. — СПб.: КОРОНА принт, 2000. — 352 с.
6. Керносов М. А. Разработка онтологии подсистемы оперативного контроля текущей успеваемости студентов / М. А. Керносов, О. С. Федоренко // Проблемы інформаційних технологій. — Херсон : ХНТУ, 2014. — № 15. — С. 177—185.
7. Лёве Дж. Создание служб Windows Communication Foundation / Дж. Лёве. — СПб. : Питер, 2008. — 592 с.
8. Мирошниченко Г. А. Реляционные базы данных: практические приемы оптимальных решений / Г. А. Мирошниченко. — СПб. : БХВ-Петербург, 2005. — 400 с.
9. Робисон У. C# / У. Робисон. — М. : ДМК Пресс, 2005. — 352 с.
10. Торрес Р. Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса / Р. Дж. Торрес. — М. : Издательский дом Вильямс, 2002. — 400 с.
11. Троелсен Э. C# и платформа .NET. Библиотека программиста / Э. Троелсен. — СПб. : Питер, 2006. — 796 с.

Serhiy USTENKO, Vladyslav PRADEDOV
Mykolaiv

SOFTWARE DEVELOPMENT FOR THE AUTOMATION OF THE ACCOUNT OF CURRENT PROGRESS OF STUDENTS

The work is dedicated to the development of software as a service-oriented application software «Electronic Journal» to automate the current progress of students. The objects of the account will be attendance, evaluation of disciplines on certain types of control (with a mark of receipt date) and indicators of readiness of students to pass the session. The e-journal will have access only to the teacher and can be filled only during class, the teacher encourages those present to celebrate and stand assessment in a timely manner, otherwise the sessions will be considered as missing and will not be counted workload.

Key words: software development, automation of accounting and analysis, the current progress of students, service-oriented applications, the service, the client.

Сергей УСТЕНКО, Владислав ПРАДЕДОВ
г. Николаев

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА ТЕКУЩЕЙ УСПЕВАЕМОСТИ СТУДЕНТОВ

Работа посвящена разработке программного обеспечения в виде сервис-ориентированного программного приложения «Электронный журнал» для автоматизации учета текущей успеваемости студентов. Объектами учета будут посещения занятий, оценки по дисциплинам по определенным видам контроля (с отметкой даты получения) и показатели готовности студентов к сдаче сессии. К электронному журналу будет иметь доступ только преподаватель и его можно будет заполнить только во время проведения занятий, что будет побуждать преподавателя отмечать присутствующих и выставлять оценки своевременно, поскольку в противном случае занятие будет считаться пропущенным и ему не будет засчитано учебную нагрузку.

Ключевые слова: разработка программного обеспечения, автоматизация учета и анализа, текущая успеваемость студентов, сервис-ориентированное приложение, служба, клиент.

Стаття надійшла до редколегії 07.03.2016

УДК 514.8

Сергій УСТЕНКО, Олександр СИНЯВІН
м. Миколаїв
ustenko.s.a@gmail.com, alexander-sinyavin@yandex.ua

ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ ПЛОСКОЇ КРИВОЇ ІЗ ПАРАБОЛІЧНОЮ КРИВИНОЮ ПРИ ЗАДАНОМУ ЇЇ ВІДХИЛЕНІ ВІД ЛІНІЙНОГО РОЗПОДІЛУ

Робота присвячена розробці нового підходу до побудови плоскої кривої лінії із параболічною кривиною, для якої задається відхилення кривини від лінійного розподілу кривини. Така задача виникає у випадках коли потрібно впливати на характер розподілу кривини ділянки плоскої кривої лінії, не змінюючи при цьому значення кривини в її граничних точках. Дослідження графіку параболічного розподілу кривини з урахуванням його відхилення від лінійного розподілу дозволило визначити залежності для обчислення невідомих коефіцієнтів параболічного та лінійного розподілів кривини. Запропонований підхід реалізовано у вигляді програмного додатку об'єктно-орієнтованою мовою програмування Object Pascal.

Ключові слова: плоска крива, кривина, розподіл кривини, геометричне моделювання, параболічний розподіл, лінійний розподіл, відхилення.

Дослідження з геометричного моделювання плоских кривих відбуваються в таких галузях: архітектурно-будівельній, раціональному розміщенні об'єктів, металообробці, сільгоспмашинобудуванні, пожежній техніці та технологіях, літакобудуванні, створенні турбін і компресорів тощо.