

**В.В. ВЛАХ**, аспірант Української академії друкарства, Львів

## МЕТОДОЛОГІЯ РЕАЛІЗАЦІЇ ЕКСПОРТУ ДАНИХ, ВІЗУАЛІЗАЦІЇ ТА ФОРМУВАННЯ 3D-МОДЕЛІ МЕХАНІЗМУ У СПЕЦІАЛІЗОВАНІЙ РОЗРАХУНКОВІЙ СИСТЕМІ

В статті розглядаються методи, за допомогою яких утворюється візуалізація механізмів та виконується передача даних з спеціалізованого програмного забезпечення в середовище AutoCAD. Пропоноване програмне забезпечення створювалось в середовищі Embarcadero Rad Studio Delphi методами об'єктно-орієнтованого програмування. Це значною мірою дало змогу реалізувати експорт кінематичної схеми механізму та сформувати код програми мовою AutoLISP, яка описує тривимірну модель.

**Ключові слова:** Delphi, об'єктно-орієнтоване програмування, AutoLISP, AutoCAD, 3D-модель.

**Вступ.** Просторові важільні механізми є важливою складовою сучасної техніки і виробничих технологій. Такі механізми застосовуються майже в усіх машинах. Конструювання таких технічних об'єктів здійснюється сьогодні з використанням САПР. Будь-який конструктор, що створює виріб з рухомими ланками, завжди хоче побачити, як він буде працювати, ще до його виготовлення. Йому важливо відстежити траєкторії руху точок, перевірити, чи не стикаються окремі частини механізму між собою чи з оточуючими елементами.

**Актуальність задачі.** Моделювання динамічних характеристик механізмів і виконавчих пристроїв машин різного призначення на стадії їх проектування є досить актуальним завданням для конструкторів. Можна стверджувати, що сьогодні існує багато видів САПР, які дозволяють проводити кінематичний та динамічний аналіз механізмів. Це переважно великі потужні системи, які дорого коштують, складні в користуванні, потребують додаткових знань та великих апаратних ресурсів робочого комп'ютера. Тому вважається, що створення простої в користуванні, багатofункціональної САПР, з високими можливостями візуалізації для кінематичного, кінетостатичного аналізу та динамічного синтезу механізмів, яка не потребувала би великих апаратних ресурсів робочого комп'ютера є актуальним.

**Аналіз останніх досліджень.** Автором статті був проведений аналіз матеріалів, які розміщені в мережі Internet. Лідер розробки таких систем Autodesk розробив безкоштовний мобільний додаток Force Effect Motion, який може бути корисним для студентів машинобудівних спеціальностей. В країнах СНД над вирішенням проблеми автоматизації розрахунку механізмів займалися в 2000-х роках, але великих результатів досягнуто не було. Тому через малу платоспроможність більшість вітчизняних вчених та інженерів використовують для розрахунків універсальні програми, такі як Excel. У Європі та США також займалися над вирішенням даного питання. Створити програмний інструмент вузької спеціалізації не вдалося, але було розроблено багато рішень, які можна інтегрувати в потужні системи для удосконалення проектування.

**Ціль статті.** Мета статті – показати методи за допомогою яких реалізовано побудову кінематичної схеми механізму, експорт даних та формування 3D-моделі механізму в створеній автором САПР для розрахунку механізмів.

© В.В. Влах, 2015

**Постановка задачі.** Пропоноване програмне забезпечення створене в середовищі Embarcadero Rad Studio Delphi [1], методами об'єктно-орієнтованого програмування та Model View Controller (MVC). Embarcadero Delphi, раніше Borland Delphi та CodeGear Delphi – це інтегроване середовище швидкої розробки програмного забезпечення.

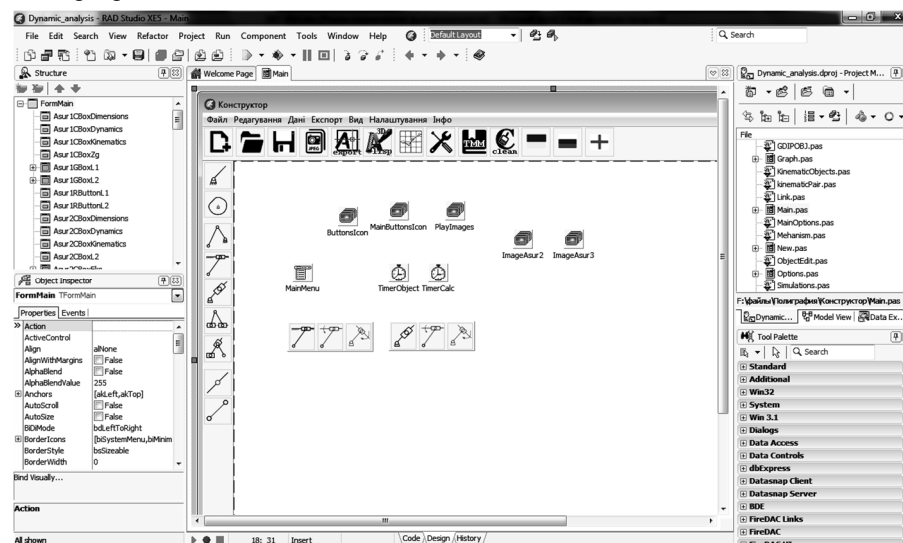


Рисунок 1 – Середовище програмування Delphi XE

Програма основана на створенні об'єктів [2], виконанні ними певних процедур та функцій, зміні своїх властивостей і обміном даних між собою. В даній роботі за основу взято класифікацію механізмів за Ассуром [3, 4]. Тобто, досліджуваний механізм завжди складається з ведучої ланки (кривошипа) і приєднаних до неї груп Ассура II класу. Кожна група Ассура, кінематична пара, ланка – це об'єкти, які зберігають в собі величини кінематичних та геометричних характеристик для подальшого розрахунку механізму. За допомогою такого методу реалізується обмін параметрами між групами Ассура та ведучою ланкою.

До складу об'єкту входить набір підпрограм, які виконуються при потребі. Такими підпрограмами є *EnterParams* (введення параметрів), *ShowParams* (показати параметри), *Calculate* (розрахнок), *KinematicScheme* (побудова кінематичної схеми), *AddPoint* (дати точку), *AddLink* (дати ланку), *SearchLink* (знайти ланку), *Searchpoint* (знайти точку), *Delete* (видалити себе зі списку об'єктів), *constructor Create* (створити себе) та інші.

Структура програми створена так, що кожен окремий об'єкт описаний в окремому модулі. Всі модулі з описами об'єктів підключаються до головного модуля програми (модуль який описує робоче середовище) і там вони вже взаємодіють між собою.

Представлена ієрархія об'єктів, структура самих об'єктів, методи їх взаємодії між собою – це все являє собою ядро програми, яке дало поштовх для реалізації багатьох нових рішень. Особливістю програми є те, що всі об'єкти, з яких формується механізм, зберігаються в дереві побудови. Це дозволяє з ле-

гкістю добратися до потрібного об'єкту і змінювати (при потребі) його структуру, записати, чи зчитати необхідні дані.

**Візуалізація механізму.** Побудова кінематичної схеми досліджуваного механізму відбувається методом послідовного виконання процедури *KinematicScheme*, яка є описана в кожному об'єкті. Тобто, якщо механізм складається з кривошипу і групи Ассура I виду, то спочатку виконується процедура побудови кінематичної схеми кривошипу, а потім групи Ассура. Дані геометричних параметрів беруться на основі розрахунків (процедура *Calculate*).

Відображення схеми на екран відбувається в кілька кроків:

- очищення віртуального растрового зображення, що створюється при запуску програми;
- розрахунок механізму;
- викреслювання схеми механізму в заданому положенні у вигляді бітової карти;
- вставка отриманого растрового зображення в робочу область головного вікна.

Анімація руху механізму відбувається за таким самим принципом, тільки за кожен наступний такт схема викреслюється в наступному положенні.

Довжини ланок механізму викреслюються в міліметрах. Для побудови схеми в віртуальній бітій карті потрібно здійснити перехід з міліметрів в пікселі певного екрану. Для того, щоб програмно визначити роздільну здатність монітора користувача, треба розрахувати, скільки розміщено пікселів в одному міліметрі на моніторі користувача. Спочатку потрібно дізнатись, скільки пікселів розміщено в одному дюймі. Для цього в програмному коді застосовано формулу:

$$pixels = \frac{mm * PPI}{25,4}$$

де PPI (англ. pixels per inch) – кількість пікселів на дюйм. Застосовується для вказівки роздільної здатності при введенні або виведенні графіки. Вимірюється числом пікселів, що припадають на дюйм поверхні.

Реалізація побудови схеми відбувається за допомогою графічного модуля GDI+ (Graphics Device Interface, Graphical Device Interface). GDI+ – це набір програмних засобів, які використовуються в Microsoft .NET. GDI+ дозволяє програмістам виводити дані на екран або на принтер без необхідності забезпечувати роботу з певними типами пристроїв відображення.

Під час виконання даної роботи було проаналізовано можливості стандартного графічного модуля GDI та GDI+. Виявилось, що реалізація методів стандартного графічного модуля Delphi не підходить для створення пропонованого програмного забезпечення. Основною причиною такого висновку стало те, що в GDI величини геометричних характеристик, на основі яких відбувається побудова кінематичної схеми, потрібно задавати цілим числом (рисунок 2). На відміну від стандартної графічної бібліотеки Delphi, GDI+ дає можливість задавати величини параметрів незалежно від типу змінної (рисунок 3).

```

Formwork.FrameDraw1.PaintBox1.Canvas.H;
end;
end;
procedure TFormwork.N10Click(Sender: TObject)
begin
if s.step > 0 then

```

Рисунок 2 – Реалізація побудови лінії в Delphi XE і типи змінних

```

DB->Image = B;
G->DrawLine(
DB->Refresh(
void DrawLine(System::Drawing::Pen ^pen, float x1, float y1, float x2, float y2)
Проводит лінію, соединяющую две точки, задаваемые парами координат.
pen: Структура System.Drawing.Pen, определяющая цвет, ширину и стиль линии.
X_1, Y_1, X_2, Y_2, X_3, Y_3, X_4, Y_4, X_5, Y_5, X_6, Y_6, X_7, Y_7;
X_1 = X - 25*cos(fi);

```

Рисунок 3 – Реалізація побудови лінії в Visual Studio і типи змінних

В результаті аналізу та порівняння було обрано методику побудови графіки на основі методів GDI+, що дозволило уникнути втрати даних при побудові кінематичної схеми. Оскільки GDI+ входить в комплект середовища програмування Visual Studio, довелося власними зусиллями знайти та підключити дану бібліотеку до системи Delphi.

**Формування 3D-моделі.** Щоб не підвищувати системні вимоги програми – тривимірна модель досліджуваного механізму будується не безпосередньо в програмі. Пропонована система автоматично формує код, який описує об'ємну модель досліджуваного механізму мовою AutoLISP для середовища AutoCAD. Код програми відображається в окремому вікні, його можна відредагувати та зберегти в форматі \*.lsp. З цього самого вікна можна відкрити AutoCAD (якщо є встановлений), який автоматично без втручання користувача побудує 3D-модель механізму. LISP-код формується за таким самим принципом як викреслювання кінематичної схеми – послідовним виконанням процедури, що відповідає за формування коду.

**Реалізація експорту даних.** В даній роботі реалізовано безпосередній експорт даних в декілька систем: Word, Excel [5] і AutoCAD [6]. Таку можливість вдалося втілити в реальність за допомогою методів Microsoft OLE Automation та COM.

OLE (англ. Object Linking and Embedding) – технологія зв'язування та впровадження об'єктів в інші документи та об'єкти, розроблена корпорацією Майкрософт. OLE дозволяє передавати частину роботи від однієї програми до іншої і повертати результати назад.

Експорт кінематичної схеми або 3D-моделі в AutoCAD здійснюється з використанням об'єктної моделі AutoCAD. Це структура об'єктів: додатки AutoCAD, його документів (відкритих креслень), об'єктів креслення (точки, лінії тощо), словників, їх властивостей і методів. Усі ці об'єкти описані в системі COM та представлені для будь-яких мов програмування. Опис об'єктної моделі здійснено у файлах з розширенням tlb (type library – бібліотека типів), а доступ до об'єктів і їх властивостями описано в реєстрі.

На рисунку 4 зображено код процедури, який виконує підключення до серверу AutoCAD а також зображено модулі, які необхідні для реалізації виконання даної процедури. Для експортування даних в Microsoft Word та Excel використовується аналогічна методика.

**Висновки.** Пропонована програма дає можливість сформувати механізм II класу довільної структури та переглянути анімацію руху досліджуваного механізму. Дерево побудови дозволяє змінювати геометричні параметри механізму, що дає змогу користувачу отримати максимально точні дані кінематичних характеристик.

Функції програми передбачають можливість дослідження спроектованого механізму в інших системах. Для цього передбачено експорт даних в офісні програми Word та Excel та систему автоматичного проектування AutoCAD.

```

unit Acad;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Data, Math,
  AutoCAD_TLB, OleServer, ExtCtrls, ComObj, ComCtrls, OleCtrls;
procedure TFormAcad.Connect;
var
  v : OleVariant;
begin
  try
    // спроба приєднатись до вже запущеної копії AutoCad
    v := GetActiveOleObject('AutoCAD.Application');
    AcadDocument1.ConnectTo((IDispatch(v) as IAcadApplication).ActiveDocument);
  except
    // якщо не вийшло приєднатись, то запускаємо нову копію AutoCad
    AcadDocument1.Connect
  end;
  // зразу заповнюємо додаткову змінну для наступної перевірки
  LastDoc:= AcadDocument1.Path+'\'+AcadDocument1.Name;
end;

```

Рисунок 4 – Код програми приєднання до серверу AutoCAD.

Експорт даних у Word та Excel дає змогу отримати стандартні документи та табличні дані для детального аналізу механізму. А експорт у середовище AutoCAD дозволяє отримати векторне зображення кінематичної схеми механізму. Однією з найбільших переваг запропонованої програми є автоматичне формування коду програми AutoLISP для побудови тривимірної моделі механізму. Отриманий код програми можна відредагувати та зберегти у файл з розширенням \*.lsp. В результаті роботи з програмою користувач отримує дані у файлах з розширеннями: \*.jpeg, \*.dwg, \*.dxf, \*.lsp, \*.xlsx, \*.docx та файл самої програми \*.kam.

Результати роботи програми можуть бути використані для подальшого кінетостатичного чи динамічного аналізів.

**Список літератури:** 1. *Архангельський А.Я.* Программирование в Delphi для Windows / А.Я. Архангельський // Верс. 2006, 2007, Turbo Delphi – СПб.: Бинном-Пресс, 2007. – 1248с. 2. *Влах В.В.* Комп'ютерний кінематичний аналіз механізмів II класу довільної структури / В.В. Влах, В.Р. Пасіка, С.М. Комаров. – Комп'ютерно-інтегровані технології: освіта, наука, виробництво, №6. – Луцьк: В-во ЛуНТУ, 2011. – С.55-58. 3. *Кіницький Я.Т.* Теорія механізмів і машин: Підручник / Я.Т. Кіницький. – К.: Наукова думка, 2002. – 660с. 4. *Пасіка В.Р.* Кінематичний аналіз циклових важільних механізмів II класу аналітичним методом проектування планів: Навч.-метод. посіб / В.Р. Пасіка. – Львів: УАД, 2009. – 68с. 5. *Чиртик А.А.* Программирование в Delphi. Триюки и эффекты / А.А. Чиртик. – СПб.: Питер, 2010. – 400с. 6. *Зувев С., Полеишук Н.* САПР на базе AutoCAD – как это делается. – БХВ-Петербург, 2004. – 1168с.

**Bibliography (transliterated):** 1. *Arkhanhel's'kyi A.Y.* Prohramyrovanye v Delphi dlya Windows / A.Ya. Arkhanhel's'kyi // Vers. 2006, 2007, Turbo Delphi – Sankt-Pererburh: Bynom-Press, 2007. – 1248p. 2. *Vlakh V.V.* Komp'yuternyy kinematychnyy analiz mekhanizmiv II klasu dovil'noyi struktury / V.V. Vlakh, V.R. Pasika, S.M. Komarov. – Komp'yuterno-intehrovani tekhnolohiyi: osvita, nauka, vyrobnytstvo, No6. – Luts'k: V-vo LuNTU, 2011. – P.55-58. 3. *Kinys's'kyi Y.T.* Teoriya mekhanizmiv i mashyn: Pidruchnyk / Ya.T. Kinys's'kyi. – Kiev: Naukova dumka, 2002. – 660p. 4. *Pasika V.R.* Kinematychnyy analiz tsyklovykh vazhil'nykh mekhanizmiv II klasu analitychnym metodom proektuvannya planiv: navch.-metod. posib / V.R. Pasika. – L'viv: UAD, 2009. – 68p. 5. *Chyrytk A.A.* Prohramyrovanye v Delphi. Tryuky y efekty / A.A. Chyrytk. – Sankt-Pererburh: Pyter, 2010. – 400p. 6. *Zuev S, Poleshchuk N.* SAPR na baze AutoCAD – kak eto delaetsya. – BKhV-Peterburh, 2004. – 1168p.

Надійшла (received) 15.05.2015