

УДК 004.415

А. В. Пархоменко, І. М. Павлюченко, О.Г. Вотінов, В. Ю. Клименко  
Запорізький національний технічний університет

## РОЗРОБКА ПРОГРАМНИХ БІБЛІОТЕК ДЛЯ САПР “КОМПАС”

© Пархоменко А. В., Павлюченко І. М., Вотінов О.Г., Клименко В. Ю., 2003

**Розглядаються можливості адаптації сучасних САПР для розв’язання конкретних задач з урахуванням специфіки підприємств. Досліджується розробка сервісних бібліотек для САПР “КОМПАС” і пропонується методика створення програмних бібліотек на основі функцій КОМПАС-API.**

**In the article capabilities of adapting modern CAD systems for solving certain tasks taking into account characteristics of enterprises are considered. Development of libraries for KOMPAS CAD is examined, and principles of creation of program libraries using KOMPAS-API functions are proposed.**

У наш час існує велика кількість різноманітних систем автоматизованого управління (АСУП), проектування (САПР), технологічної підготовки виробництва (АСУТП). Головним користувачем САПР та АСУТП закономірно виступає промисловість, у той час як АСУП знаходять застосування у більшому спектрі організацій. Для розвитку сучасних автоматизованих систем характерні дві основні тенденції: розширення сфери використання та спеціалізація для більш зручного їх застосування. Ці дві тенденції протистоять одна одній, а популярна САПР має одночасно і бути універсальною, і надавати певні спеціалізовані можливості для кожної з областей використання.

Вирішити це протиріччя покликана відкрита архітектура сучасних САПР, що дозволяє користувачам додавати до системи свої спеціалізовані модулі, які використовують універсальні функції системи. У САПР AutoCAD (розробник “AutoDESK”, США) для такої мети служить AutoLISP, або може використовуватись Visual Basic. Для САПР КОМПАС (розробник АТ “АСКОН”) існують засоби розробки додатків КОМПАС-МАЙСТЕР, які дозволяють використовувати мови програмування C++ Builder, Delphi, MS Visual C++, Borland C++. Формат програмних бібліотек КОМПАС \*.rtw сумісний зі стандартним форматом бібліотек DLL [1].

Для багаторазового використання однакових елементів і побудов у системі КОМПАС передбачені крім програмних бібліотек ще два варіанти бібліотек користувача. Залежно від того, з якими змінами використовуватиметься елемент надалі, можна обрати один із видів представлення: параметричний чи непараметричний фрагмент, або функцію програмної бібліотеки (рис. 1).

Три типи бібліотек користувача, які можуть бути створені і використані у системі КОМПАС, призначені для різних цілей, мають різні можливості, недоліки і переваги.

Бібліотеки непараметризованих фрагментів дозволяють лише зберегти елементи у тому вигляді, у якому вони існують на момент створення бібліотеки. При доданні на креслення фрагментів із бібліотеки неможливо буде довільно змінювати їх параметри. Вони можуть використовуватись для збереження елементів з незмінними розмірами, наприклад, елементів принципівих схем. Прикладом може бути бібліотека “Условные обозначения Пневмо- и Гидросхем”.

Бібліотеки параметризованих фрагментів дозволяють зберегти образ елемента та математичні зв'язки між геометричними розмірами різних його частин. Тоді при доданні фрагментів на креслення і їх зміні разом з одним розміром елемента будуть змінюватись усі пов'язані з ним розміри.

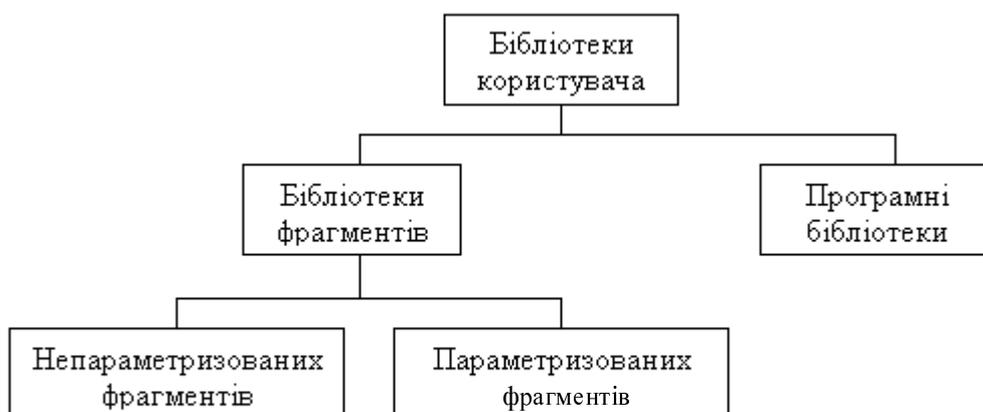


Рис. 1. Види бібліотек САПР КОМПАС

Вони можуть використовуватися для збереження елементів зі змінними параметрами, що пов'язані між собою явними залежностями. Прикладом тут може бути бібліотека “Типовые элементы”.

Але є багато призначень, для яких бібліотеки фрагментів неприйнятні. У першу чергу це побудови, що потребують складних циклічних розрахунків або якогось вибору. Крім того, лише у програмній бібліотеці можна організувати розгорнутий діалог з користувачем, надавати йому підказки і т. ін. Тому програмні бібліотеки є найвищим ступенем автоматизації на цьому рівні. Вони дозволяють по заданій програмі побудувати створюваний елемент згідно з заданими командами.

Бібліотеки програмних функцій є найбільш універсальними. Вони можуть з успіхом замінювати два попередні типи бібліотек, але це може іноді потребувати більших витрат часу на їх розробку. На цей час розроблена велика кількість бібліотек, що розширюють можливості системи КОМПАС. Ці бібліотеки розроблені як АТ “АСКОН”, так і користувачами системи, наприклад, бібліотека “САПР ФРЕЗ” (автоматизована розробка черв'ячних фрез) та інші. Бібліотеки, розроблені користувачами, широко використовуються, що є ознакою того, що існуючі засоби розробки програмних бібліотек достатні для створення повноцінного якісного продукту [2], [3].

Метою цієї роботи є дослідження можливостей, що надаються бібліотеками \*.rtw.

Як свідчать проведені дослідження, до складу КОМПАС-МАЙСТЕР входять бібліотеки функцій-прототипів на відповідних мовах програмування, що забезпечують зв'язок програмних бібліотек з функціями системи КОМПАС. Бібліотеки функцій-прототипів декларують функції КОМПАС, які дозволяють взаємодіяти з САПР — додавати у документ графічні елементи та редагувати їх, записувати або зчитувати дані з файлів. Серед інших можливостей функцій КОМПАС можна зазначити методи знаходження рішень різноманітних геометричних задач.

Для прикладу розглядається розробка бібліотеки в середовищі Microsoft Visual C++ 6.0. Щоб створювана програмна бібліотека могла взаємодіяти із САПР КОМПАС, треба підключити до неї файли `arrautox.cpp`, `arrautox.h` і деякі інші залежно від функцій, які будуть використані у бібліотеці.

У бібліотеці розробник має описати декілька необхідних функцій: `LIBRARYENTRY`, `LIBRARYNAME`, `LIBRARYID`. Головною серед них є `LIBRARYENTRY`, яка обробляє команди, що надходять від інтерфейсу КОМПАС. Функція `LIBRARYNAME` повертає назву бібліотеки, а `LIBRARYID` — її ідентифікатор.

Перед тим як перейти до програмування бібліотеки, слід зупинитися на загальній схемі роботи системи КОМПАС із бібліотеками. Схема взаємодії програмної бібліотеки з КОМПАС наведена на рис. 2.

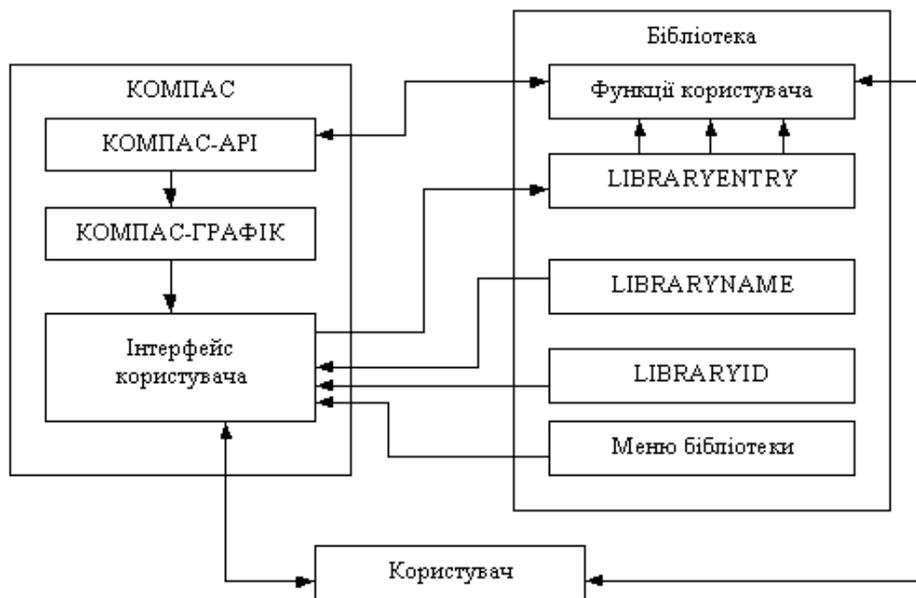


Рис. 2. Схема взаємодії програмної бібліотеки з САПР КОМПАС

Взаємодія виконується через діалогове вікно (Інтерфейс користувача), яке має назву бібліотеки і меню команд, які можуть бути виконані в системі КОМПАС.

Залежно від вибору користувачем пункту з меню бібліотеки, інтерфейс КОМПАС передає ідентифікатор обраного пункту до функції `LIBRARYENTRY` як номер команди. Ця функція повинна залежно від отриманого номера виконати дії, що передбачені обраним пунктом меню. Зазвичай це оформлюється як виклик відповідної функції з бібліотеки, яка виконує необхідні дії. При цьому функція з бібліотеки звертається до функцій `КОМПАС-API`, а також може взаємодіяти безпосередньо з користувачем. Функції `КОМПАС-API` виконують доступ до графічної бази даних системи `КОМПАС-ГРАФІК` для формування і обробки графічних документів, потім бібліотека знову переходить у режим очікування команди.

При розробці бібліотеки функцій для САПР КОМПАС зручніше користуватись методом проектування зверху вниз. Методика розробки бібліотеки складається з таких етапів.

1. Розробити структуру меню бібліотеки, виходячи з цілей, які вона має виконувати. Меню є важливою частиною бібліотеки, оскільки команда до функції `LIBRARYENTRY`

надходить саме у результаті вибору якого-небудь пункту меню. Меню бібліотеки розробляється звичайними засобами MS Visual C++.

2. Взяти за основу програми каркас вже існуючої бібліотеки, де описані всі дії з ініціалізації необхідних змінних, а також підключені всі потрібні заголовочні файли. У якості такої базової бібліотеки можна порекомендувати КОМПАС\AppTools\C++\VISUALC\step2\ зі стандартної поставки КОМПАС. Ця бібліотека виконує всі необхідні дії з підготовки до роботи і вже має описану функцію LIBRARYENTRY та меню бібліотеки. Треба вилучити всі непотрібні функції, які наводяться у бібліотеці для прикладів, але це простіше, ніж створення бібліотеки “з нуля”.

Функція LIBRARYENTRY складається, головним чином, з “перемикача” switch, у якому залежно від команди, що прийшла від інтерфейсу САПР через аргумент цієї функції, викликаються всі інші функції бібліотеки, які реалізують її конкретні можливості.

У MS Visual C++ потрібно на початку роботи функції LIBRARYENTRY викликати функцію GetStaticModuleState для створення можливості використовувати ресурси, що описані у бібліотеці. Потім треба виконати послідовність дій для забезпечення зв'язку функцій бібліотеки з об'єктами КОМПАС-API, яку розробники пропонують виділяти в окрему функцію GetKompas. Ця функція ініціалізує вказівник kompas на інтерфейс KompasObject, за допомогою якого і виконується зв'язок з КОМПАС. Текст цієї функції можна знайти у будь-якому прикладі з тих, що надаються у стандартній поставці КОМПАС.

3. На основі каркасу (п. 2), розробити функцію LIBRARYENTRY, через яку реалізується виклик функцій користувача. Значення параметра com в ній дорівнює ідентифікатору відповідного пункту меню.

4. Реалізувати функції користувача і приєднати їх до відповідних пунктів меню через функцію LIBRARYENTRY. Тобто, подальша розробка бібліотеки складається з реалізації конкретних функцій за бажанням користувача. Треба зазначити, що у деяких бібліотеках вибір дії користувачем виконується у діалогових вікнах, а меню бібліотеки складається з одного пункту, який викликає початкове діалогове вікно.

Розглянемо далі групи функцій КОМПАС-API, які будуть необхідні у більшості бібліотек. Вони реалізовані для MS Visual C++ з використанням сучасної технології Automation через інтерфейси IDispatch.

Дві важливі групи функцій можна побачити у інтерфейсах ksDocument2D та ksMathematic2D. Вони дозволяють, відповідно, працювати з графічними елементами та виконувати математичні обчислення для двомірних креслень. Методи інтерфейсу ksDocument2D дозволяють будувати графічні примітиви, об'єднувати їх у групи, редагувати або видаляти, перетворювати системи координат, будувати спеціальні види, ставити розміри тощо, а також записувати або зчитувати креслення з диску. Методи інтерфейсу ksMathematic2D дають можливість розраховувати точки перетину прямих і кривих, будувати дотичні та спряження, знаходити відстані від точки до прямої або кривої і таке інше.

Окремо виділяється інтерфейс ksIterator, який є необхідним для знаходження вказівників на вже існуючі графічні елементи.

Існують також функції взаємодії з користувачем, що дозволяють йому ввести які-небудь числові або строкові параметри безпосередньо під час виконання функції бібліотеки користувача, але вони є допоміжними на відміну від інтерфейсу, побудованого у програмному середовищі MS VC++.

Детальніший опис цих та інших функцій можна отримати у довідковій системі КОМПАС-МАЙСТЕР у файлі apptools.hlp.

Розглянемо докладніше основні можливості, які надає програмісту КОМПАС-API. При розробці багатьох бібліотек функцій для САПР КОМПАС програміст досить часто стикається з необхідністю додати на креслення графічний елемент, запропонувати користувачеві ввести будь-яке значення параметра або знайти серед об'єктів на кресленні один конкретний і певним чином змінити його. Також можуть виникати математичні задачі, наприклад, знаходження перетину двох ліній.

Для роботи з кресленням та математикою потрібно отримати вказівники на екземпляри інтерфейсів ksDocument2D та ksMathematic2D:

```
ksDocument2D doc( Kompas.ActiveDocument2D() );
```

```
ksMathematic2D mat( Kompas.GetMathematic2D() );
```

Після цього для побудови, наприклад, відрізка лінії, потрібно викликати `doc.ksLineSeg( 0, 40, 100, 40, 1 );`

Ця функція будує лінію від точки (0;40) до (100;40).

Для знаходження точок перетину двох ліній треба:

```
//створити масив для координат точок
```

```
ksDynamicArray Points( Kompas.GetDynamicArray(POINT_ARR));
```

```
//отримати ці координати
```

```
mat.ksIntersectLinLin(10, 10, 0, 15, 5, 90, Points);
```

Ці лінії проходять через точки (10; 10) та (15; 5) під кутами  $0^\circ$  та  $90^\circ$  відповідно. Результат — точка перетину ліній, яка знаходиться у нульовому елементі масиву Points, який також є об'єктом КОМПАС-API. Для звернення до нього треба викликати метод `ksGetArrayItem`:

```
ksMathPointParam P( Kompas.GetParamStruct(ko_MathPointParam) );
```

```
Points.ksGetArrayItem(0, P);
```

Для отримання від користувача якого-небудь числового значення у змінну `x` можна викликати:

```
int x;
```

```
Kompas.ksReadInt ("Введіть число від 0 до 100", 0, 100, &x);
```

Для того, щоб пройти по всіх об'єктах поточного документа, треба виконати наступні дії:

```
reference obj;
```

```
ksIterator iter( Kompas.GetIterator() );// створити ітератор
```

```
if ( !iter.m_lpDispatch )
```

```
return;
```

```
if ( iter.ksCreateIterator( ALL_OBJ,0 ) ){ //ініціалізувати ітератор
```

```
if ( doc.ksExistObj( obj = iter.ksMoveIterator( "F" ) ) ){
```

```
//перейти на перший об'єкт, якщо він існує
```

```
do {
```

```
doc.ksLightObj( obj, 1 );//обробити об'єкт (у даному випадку — підсвітити його)
```

```
} while( doc.ksExistObj(obj=iter.ksMoveIterator("N"));
```

```
//перейти на наступний об'єкт, якщо він існує
```

```
}
```

```
iter.ksDeleteIterator();//знищити ітератор
```

```
}
```

Ці приклади мали на меті показати базові можливості КОМПАС-АРІ для виконання завдань, що постають при реалізації більшості проектів. Вони демонструють деяку частину того інструментарію, який стає доступним для програміста у взаємодії з САПР КОМПАС.

У результаті проведених досліджень можна відзначити широкий спектр задач, які можуть бути розв'язані за допомогою програмних бібліотек САПР КОМПАС. При цьому не накладається ніяких обмежень на складність розрахунків, а також на способи взаємодії з користувачем. КОМПАС-АРІ надає ряд функцій для роботи з документами КОМПАС, і дозволяє при цьому використовувати всі можливості сучасних мов програмування.

*1. Ткаченко В.П., Воронцов Б.С., Бочарова И.А., Павлюченко И.Н. Компьютерная графика для инженера: Учебн. пособие. — Луганск: Изд-во СЛУ им. В. Даля, 2002.—148 с. 2. КОМПАС — ГРАФИК 5. X.: Практическое руководство. Ч. 1.— АО АСКОН, 1999. —468 с. 3. КОМПАС–ГРАФИК 5. X. Руководство пользователя. Ч. 1. — АО АСКОН, 2001.— 473 с.*

УДК 621.313

**Л.В. Каша**

Національний університет “Львівська політехніка”,  
кафедра ЕМА

## **АВТОМАТИЗОВАНА СИСТЕМА ДОСЛІДЖЕННЯ ЕЛЕКТРОПРИВОДІВ НА БАЗІ ВЕНТИЛЬНИХ РЕАКТИВНИХ ДВИГУНІВ**

© Каша Л.В., 2003

**Описано автоматизовану систему дослідження електроприводів на базі вентильних реактивних двигунів.**

**The automated research system of electric drives on the basis of switched reluctance motors are given in this paper.**

Високі темпи науково-технічного прогресу, а також перехід до ринкової економіки в Україні вимагають неперервного збільшення обсягу знань, якими повинні володіти спеціалісти в галузі проектування та дослідження електричних машин. На зламі тисячоліть невід'ємною частиною освіти спеціалістів стає застосування інформаційних технологій. Комп'ютерна техніка незамінна при накопиченні “баз знань”, автоматизації проектно-конструкторських розрахунків, діагностуванні електричних машин та дослідженні процесів, що проходять в них.

Вентильний реактивний двигун з буферами енергії належить до ще мало вивчених електричних машин, а тому створення досконалих схем і конструкцій, а також успішне вирішення задач його аналізу та синтезу залежать від наявності адекватних математичних моделей електромеханічних процесів в ньому.