

Методика разработки прикладного программного обеспечения в CAD/CAM

Л. Мегрелишвили

CAD/CAM - это интегрированная система компьютерного проектирования решающая комплекс задач технологического назначения.

На сегодняшний день существует два типа систем CAD/CAM - систем, это системы с закрытой архитектурой и системы с открытой архитектурой. Системы с закрытой архитек-

ЗАКРЫТАЯ АРХИТЕКТУРА



ПРОГРАММНЫЕ МОДУЛИ
СОЗДАЮТСЯ РАЗРАБОТЧИКОМ

турой решают узкий класс задач пользователя, системы с открытой архитектурой строятся на базе экспертных систем, что позволяет пользователю самому создавать собственные CAD/CAM- системы.

Разработка CAD/CAM - систем с открытой архитектурой осуществляется в двух укрупненных направлениях:

- Разработка экспертной системы которая реализует метафункции и разрабатывается разработчиком.

ОТКРЫТАЯ АРХИТЕКТУРА



ПРОГРАММНЫЕ МОДУЛИ
СОЗДАЮТСЯ ПОЛЬЗОВАТЕЛЕМ

- Разработка собственной CAD/CAM - системы которая решает конкретные задачи пользователя и разрабатывается пользователем.

Создание собственной CAD/CAM - системы осуществляется посредством создания собственных программных модулей функционирующих в среде экспертной системы.

Целью данного исследования было разработка методики создания программных модулей. В данном докладе рассмотрена методика создания программных модулей для формирования Управляющих Программ (УП) перемещения инструмента в среде экспертной системы **Turbo T**. Целью разработки данной методики является блочно - модульная

структура программирования.

Структурированность определяет правило организации внутри программы связей по передаче данных и управления. Структурированность упрощает и уменьшает объем программы, увеличивает скорость компиляции и интерпретации, повышает эффективность и надежность программирования.

Модуль разделен на отдельные структурные единицы, блоки которые физически реализованы внутри программы, функционируют независимо друг от друга и выполняют часть общей задачи, в частном случае блоку может соответствовать и отдельный программный модуль.

В результате анализа были выделены следующие шесть структурных единиц - блоков:

Первый блок - блок анализа, проверяет и присваивает текущие значения внутренним параметрам.

Второй блок - блок вычисления, вычисляет приращения к опорным точкам для определения опорных точек эквидистанты, число проходов, устанавливает признак конца зоны.

Третий блок - реализует типовые перемещения инструмента.

Четвертый блок - задает перемещение инструмента по эквидистанте и непосредственно по контуру детали.

Пятый блок - блок буфера, запоминает промежуточные результаты.

Шестой блок - блок коррекции режимов обработки.

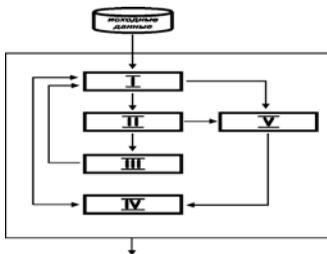
С учетом приведенной выше методике был разработан программный модуль СИКЛ L95 формирующий траекторию перемещения инструмента для многопроходной обработки типовых деталей типа втулка и фланец в среде экспертной системы **Turbo T**.

Входными данными первого блока являются характеристики описывающие контур детали, глубину резания, и припуск под полустовую обработку. Анализируя корректность задания, входные данные

I	Анализа
II	Вычисления
III	Реализации типовых перемещений инструмента
IV	Перемещения по эквидистанте
V	Буфера
VI	Коррекции режима обработки

присваивает их внутренним параметрам.

Функциями второго блока является вычисления приращений к опорным точкам, определение количества проходов и глубины резания, устанавливаются также ограничения по перемещению инструмента и устанавливается признак конца зона, вычисляются также значения центральной точки для окружности, в случае, если перемещение в зоне по окружности. Последним этапом функционирования второго блока является передача определенных значений третьему блоку и помещение некоторых своих значений в блок буфер.



Третий блок реализует перемещение инструмента по схеме петля. Выполнив определенное количество проходов в текущей зоне, передает управление первому блоку который передает характеристики следующей зоны, для обработки второму блоку. Завершив обработку существующего количества зон управление передается первому блоку который устанавливает завершение многопроходной обработки и передает данные и управление четвертому блоку.

Четвертый блок получая значения координат опорных точек от первого блока, приращения для определения опорных точек эквидистанты и значения центральной точки в случае перемещения по окружности, задает перемещение по эквидистанте. После завершения перемещения по эквидистанте, осуществляется перемещение непосредственно по контуру, после завершения перемещения по контуру инструмент возвращается в исходную точку.

Пятый блок хранит результаты вычислений второго блока, приращения к опорным точкам контура и значения центральной точки, и передает их четвертому блоку для осуществления полуступенчатой обработки детали. В режиме компиляции мы получаем инструкции *CLDATA* для многопроходной обработки деталей указанного выше класса.