

## ПРОБЛЕМЫ ПРОЦЕДУРНОГО ПОДХОДА В ПРОЕКТИРОВАНИИ ЭЛЕКТРИЧЕСКИХ МАШИН

*Розглянуті питання проектування електричних машин традиційним методом, що характеризується процедурним підходом до організації проекту і альтернативного методу, що ґрунтується на концепції об'єктно-орієнтованого проектування. Доведені переваги об'єктно-орієнтованого підходу при розбитті електричної машини на об'єкти, що включають в себе дані і функціональні модулі.*

*Рассмотрены вопросы проектирования электрических машин традиционным методом, характеризующимся процедурным подходом к организации проекта и альтернативного метода, основывающегося на концепции объектно-ориентированного проектирования. Доказаны преимущества объектно-ориентированного подхода при разбиении электрической машины на объекты, включающие в себя данные и функциональные модули.*

Одной из особенностей живой природы является ее способность порождать потомство, обладающее характеристиками, сходными с характеристиками предыдущего поколения. Природе потребовалось миллиарды лет, чтобы достичь современных форм жизни, и результат этих усилий передается каждому представителю отдельного вида живых организмов. Это результат наследования.

Если наследование приводит к таким результатам в природе, то почему оно не может быть полезно в проектировании? Электрические машины можно разбить на классы, соответствующие принципам формирования конкретных видов машин из обобщенных, обладающих универсальными для порожденных классов характеристиками [1]. Наследование позволяет, практически без ограничений, последовательно строить и расширять созданные классы. Каждый раз, когда из предыдущего класса производится последующий, производный класс наследует какие-то или все родительские качества, добавляя к ним новые. Завершенный проект может включать десятки и сотни классов, но все они могут быть произведены от считанного количества базовых классов.

Порождая классы от базовых, можно эффективно использовать алгоритмы базового класса для собственных нужд. Концепция имеет параллель в живой природе: ДНК можно рассматривать как базовый материал, из которого может быть создано любое существо. Каждый организм повторно использует ДНК для воспроизведения своего собственного вида. В проектировании повторное использование осуществляется через наследование.

Применительно к электромеханическим структурам, в качестве базового материала можно рассматривать электромеханические хромосомы, представляющие собой различные конфигурации электромагнитных полей. Их дальнейшее развитие определяет, какой вид электрической машины будет создан [2-4].

Традиционное проектирование сопровождается процедурным подходом к решению задач расчета электрической машины. Зачастую, например, в курсовом проектировании, расчеты разбиваются на последовательно выполняемые пункты с набором формул. Проект, построенный на основе процедурного метода, разделяется на функциональные модули, каждый из которых в идеальном случае выполняет некоторую законченную последовательность действий и имеет явно выраженные связи с другими функциональными модулями программы.

Можно развить идею разбиения проекта на

функциональные модули, объединив несколько функциональных модулей в один блок (иногда блок представляет отдельный проект). При этом сохраняется процедурный принцип: проект делится на несколько компонентов, каждый из которых представляет собой набор инструкций.

Деление проекта на функциональные модули и блоки является основой традиционного структурного проектирования.

Однако, как бы эффективно ни применялся структурный подход, он не позволяет в достаточной степени упростить большие и сложные проекты. Существует две основные проблемы процедурного подхода. Первая заключается в неограниченности доступа функциональных модулей проекта к глобальным данным. Вторая состоит в том, что разделение данных и методов их обработки, являющееся основой структурного подхода, неадекватно отражает картину реального мира [5].

Рассмотрим пример неконтролируемого доступа данных на примере программы проектирования асинхронного двигателя с короткозамкнутым ротором. В проекте выделим следующие функциональные модули:

- 1) выбор главных размеров;
- 2) определение числа зубцов  $Z_1$ , числа витков  $W_1$  и сечения провода обмотки статора;
- 3) расчет размеров зубцовой зоны статора и воздушного зазора;
- 4) расчет ротора;
- 5) расчет намагничивающего тока;
- 6) параметры рабочего режима;
- 7) расчет потерь;
- 8) расчет рабочих характеристик;
- 9) расчет пусковых характеристик;
- 10) тепловой расчет.

В процедурной программе, написанной, к примеру, на языке C++, существует два типа данных. Локальные данные находятся внутри функциональных модулей и предназначены для использования исключительно этими функциональными модулями. Например, в функциональном модуле п.6 "Параметры рабочего режима" используются локальные данные коэффициентов пазового, лобового и дифференциального рассеяния. Локальные данные этого функционального модуля недоступны никому, кроме самого модуля, и не могут быть изменены другими модулями.

Если существует необходимость совместного использования одних и тех же данных несколькими функциями, то данные должны быть объявлены как глобальные. Это, как правило, касается тех данных

проекта, которые являются наиболее важными. Любой функциональный модуль имеет доступ к глобальным данным. Схема, иллюстрирующая концепцию локальных и глобальных данных, приведена на рис. 1.



Рис. 1. Глобальные и локальные данные

Примером глобальных переменных могут служить мощность, частота вращения, токи, активные и индуктивные сопротивления обмоток статора и ротора, число зубцов и витков обмоток и т.д.

Большие программы обычно содержат множество функциональных модулей и глобальных переменных. Проблема процедурного подхода заключается в том, что число возможных связей между глобальными переменными и функциональными модулями может быть очень велико, как показано на рис. 2.



Рис. 2. Процедурный подход

Большое число связей между функциями и данными, в свою очередь, также порождает несколько проблем. Во-первых, усложняется структура программы. Во-вторых, в программу становится трудно вносить изменения. Изменение структуры глобальных данных может потребовать переписывания всех функциональных модулей, работающих с этими данными. Например, если разработчик программы решит записывать сопротивление в комплексной форме вида  $Z = R + jX$ , то будет необходимо внести новый тип данных, поддерживающий операции с комплексными числами. Кроме того, токи также нужно будет записать в комплексной форме. Это означает, что во все функциональные модули, оперирующие с сопротивлениями и токами, должны быть внесены изменения, позволяющие обрабатывать данные в комплексной форме. Можно привести аналогичный бытовой пример, когда в супермаркете изменяется расположение отделов, и покупателям приходится соответствующим образом менять свой привычный путь от одного отдела к другому.

Когда изменения вносятся в глобальные данные больших программ, бывает непросто определить, какие функциональные модули необходимо скорректировать. Даже в том случае, когда это удается сделать, из-за многочисленных связей между функциональными модулями и данными исправленные модули начинают некорректно работать с другими глобальными данными. Таким образом, любое изменение влечет за собой далеко идущие последствия.

Вторая, более важная проблема процедурного

подхода заключается в том, что отделение данных от расчета в функциональных модулях оказывается малоприменимым для отображения картин реального мира. В реальном мире нам приходится иметь дело с физическими объектами, такими, например, как люди и электрические машины. Этот объект нельзя отнести ни к данным, ни к функциональным методам, поскольку реальные вещи представляют совокупность свойств и поведения.

Примерами свойств (характеристик) для людей могут являться цвет глаз или место работы; для электрических машин – мощность двигателя и количество зубцов статора. Таким образом, свойства объектов равносильны данным в программах: они имеют определенное значение, например, голубой для цвета глаз или 36 для числа зубцов статора.

Поведение – это некоторая реакция объекта в ответ на внешнее воздействие. Например, человек на просьбу о помощи может дать ответ "да" или "нет". Если ввести активное сопротивление в цепь ротора асинхронного двигателя с фазным ротором, то произойдет увеличение критического скольжения и, как следствие, торможение двигателя. Ответ и торможение являются примерами поведения. Поведение сходно с функциональными модулями: функциональный модуль вызывается, чтобы совершить какое-либо действие (например, рассчитать величину пускового тока), и функциональный модуль совершает это действие.

Таким образом, ни отдельно взятые данные, ни отдельно взятые функции не способны адекватно отобразить объекты реального мира.

Основополагающей идеей объектно-ориентированного подхода является объединение данных и действий, производимых над этими данными, в единое целое, которое называется объектом.

Функции объекта, или функциональные модули, предназначены для доступа к данным объекта. Если необходимо получить какие-либо данные объекта, нужно вызвать соответствующий метод, который выполнит считывание и возвратит требуемое значение. Прямой доступ к данным извне невозможен. Данные объекта скрыты от внешнего воздействия, что защищает их от случайного изменения.

Говоря в терминах языка программирования C++, данные и методы объекта инкапсулированы.

Если необходимо изменить данные объекта, то, очевидно, это действие также будет возложено на методы объекта. Никакие другие функции не могут изменять данные объекта. Такой подход облегчает написание, редактирование и использование проекта. При этом, типичный проект состоит из совокупности объектов, взаимодействующих между собой посредством вызова методов друг друга. Структурная схема объектно-ориентированного проекта представлена на рис. 3.

При подходе к решению задачи с помощью объектно-ориентированного метода, вместо проблемы разбиения задачи на функции, ставится вопрос разбиения этой задачи на объекты. Мышление в терминах объектов оказывается более простым и наглядным, чем в терминах функций, поскольку программные объекты схожи с объектами реального мира. Приведем пример таких объектов:

- Физические объекты – статор, ротор, обмотка, электродвигатель.
- Параметрические объекты – комплексные числа, массивы, время.

- Графические объекты – линии, прямоугольники, круги.

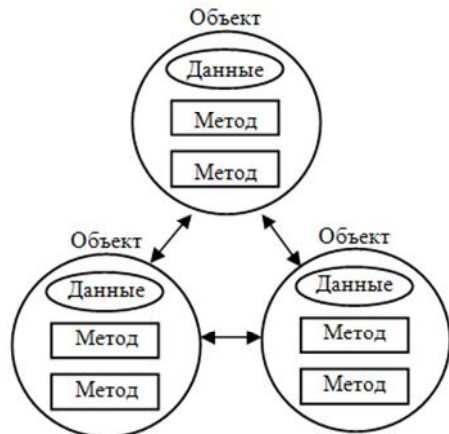


Рис 3. Объектно-ориентированный подход

Соответствие между программными и реальными объектами является следствием объединения данных и функций. Ни одна программная модель, созданная на основе процедурного подхода, не способна отразить существующие вещи столь точно, как это удается сделать с помощью объектов.

Когда идет речь об объектах, то имеется в виду, что они являются экземплярами классов. Это означает, что класс является своего рода формой, определяющей, какие данные и функции будут включены в объект класса. При формировании класса не создаются никакие объекты этого класса, по аналогии с тем, что существование электрической машины на чертеже еще не означает наличие изготовленной по ним машины. Таким образом, класс является описанием совокупности сходных между собой объектов. Например, синхронные генераторы и асинхронные двигатели относятся к классу машин переменного тока. Не существует конкретной машины, в которой вместо серийного номера было бы написано "машина переменного тока", однако машины со своими уникальными серийными номерами являются объектами этого класса, если они обладают определенным набором характеристик. Поэтому объект класса и называется экземпляром класса. В проектировании объекты, обладающие всеми признаками своих классов, создаются при выполнении расчетной программы, когда под них физически выделяется память.

Объектно-ориентированное проектирование никак не связано с процессом выполнения проекта и расчетами, а является лишь способом его организации.

Разработанные классы могут повторно использоваться в других проектах, реализуя концепцию повторного использования. Проектант может взять существующий класс, и, ничего не изменяя, внести в него свои элементы. Все производные классы унаследуют эти изменения, и в то же время каждый из производных классов можно модифицировать отдельно.

Легкость повторного использования является важным достоинством объектно-ориентированного проектирования.

#### ВЫВОДЫ

- 1) Традиционное структурное проектирование с процедурным подходом сопровождается двумя основными проблемами:
  - а) увеличение, с ростом сложности проекта, чис-

ла связей между глобальными данными и расчетными функциональными модулями;

б) отделение данных от функциональных модулей непригодно для отображения объектов реального мира, обладающих свойствами и поведением.

2) Наличие проблем процедурного подхода затрудняет изменение и сопровождение проекта, делает неосуществимой концепцию наследования.

3) Совмещение данных и функциональных модулей в одном объекте адекватно отображает свойства и поведение объектов реального мира.

4) Решением проблем процедурного проектирования является переход к объектно-ориентированному проектированию, являющемуся основой наследования, полиморфизма и повторного использования.

#### СПИСОК ЛИТЕРАТУРЫ

1. Заблодский Н.Н., Шинкаренко В.Ф., Плюгин В.Е., Гринь Г.М. Объектно-ориентированное проектирование электротепломеханических преобразователей // Техн. электродинамика. – К.: Ін-т електродинаміки НАН України. – Ч.3. – 2008. – С. 106-112.
2. N.N. Zablodsky, V.E. Plyugin etc. Object oriented designing of electro-thermo-mechanical converters with optimum thermodynamic structure // A Dunajvarosi Foiskola Kozlemeyei, № XXIX/1. – 2007. – P. 193-200.
3. Шинкаренко В.Ф. Основы теорії еволюції електромеханічних систем. – К.: Наукова думка, 2002. – 288 с.
4. V.F. Shinkarenko, N.N. Zablodsky, V.E. Plyugin, G.M. Grin. The macro-genetic analysis and systematization of specific variety of electro-thermo-mechanical converters of energy // A Dunajvarosi Foiskola Kozlemeyei, № XXIX/1. – 2007. – P. 185-192.
5. Р. Лафоре. Объектно-ориентированное программирование в C++. – СПб. – 2004. – 923с.

Поступила 17.09.2010

*Шинкаренко Василий Федорович, д.т.н., проф.*  
Национальный технический университет Украины  
"Киевский политехнический институт"  
кафедра электромеханики  
03056, Киев, пр. Перемоги, 37  
e-mail: svf46@voliacable.com

*Заблодский Николай Николаевич, д.т.н., проф.*  
*Плюгин Владислав Евгеньевич, к.т.н., доц.*  
Донбасский государственный технический университет  
кафедра "Электрические машины и аппараты"  
94204, Алчевск, пр. Ленина, 16  
тел. (06442) 2-68-87, e-mail: rector@dgmi.al.lg.ua  
тел. (06442) 3-82-66, e-mail: vlad.plyugin@ukr.net

*Мухаммад Тарик, к.фил.н.,*  
COMSATS институт информации и технологии  
Пакистан, Юниверсити-роуд, Тоб Кемп,  
Абботтабад, К. Пухтун Хва  
Департамент экологии  
e-mail: tariqjalal61@hotmail.com

*N.N. Zablodsky, V.F. Shinkarenko, V.E. Plyugin, Muhammad T.*  
**Problems of procedural approach in electric machine designing.**

Problems of electric machine designing with a traditional method, which is characterized by a procedure-oriented algorithm in project organization, and an alternative method, which is based on the object-oriented design conception, are considered. Advantages of the object-oriented method are proven under partition of an electric machine into objects that include data and functional modules.

**Key words – object-oriented, designing, electric machine, class, object, functional module, inheritance, structure.**