

УДК 60.011.55

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ: ОСНОВНЫЕ ПОНЯТИЯ И НАЗНАЧЕНИЕ

А.Ю. Сикиржицкий^а

Аннотация

В статье раскрываются понятия модульного программирования как одного из особых способов разработки программ, рассматриваются плюсы и минусы данного подхода, назначение, приводится пример модульности при разработке системы «Управление закупками».

Ключевые слова: модульное программирование, модуль, модульность в языках программирования, программы.

MODULAR PROGRAMMING: BASIC CONCEPTS AND PURPOSE

A.Y. Sikirzhytski^a

Abstract

The article describes the concept of modular programming as one of the ways to develop special programs, examines the pros and cons of this approach, purpose, an example of modularity in the design of the «Procurement Management».

Keywords: modular programming, module, modular programming languages program.

Введение

При разработке современного программного обеспечения существуют и используются множество различных методов, один из которых нам известен под именем «программирование сверху-вниз» или «пошаговая детализация». Основная идея этого метода проста: первоначальная задача разделяется на достаточно большие и относительно обособленные друг от друга подзадачи. В том случае, когда полученные подзадачи могут быть достаточно легко реализованы, для каждой из них разрабатывается соответствующий алгоритм, иначе следует дальнейшее разбиение на более простые подзадачи и т.д. Далее следует реализация каждой полученной задачи на каком-либо языке программирования, и такая реализация обычно называется (программным) модулем. Получается, что использование модулей и модульного подхода является естественным способом разработки и построения сложных программ. Стоит отметить, что и сам модуль допускает дальнейшую пошаговую детализацию, однако полученные подзадачи реализуются уже в виде процедур и функций [1].

Модульное программирование является одним из основных способов разработки программного обеспечения в наши дни, которое строится из нескольких относительно независимых друг от друга частей – модулей. Значимость модульности является одним из основополагающих при разработке современного программного обеспечения.

Модуль — функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части предназначенный для использования в других программах. Модули позволяют разбивать сложные задачи на более мелкие в соответствии с принципом модульности.

^а Сикиржицкий Алексей Юрьевич,
Минский институт управления,
магистрант,
Sikirzhytski Alexey Yurievich,
Minsk Institute of Management,
Master's degree student,
mastak@tut.by

Модульность в языках программирования — это принцип, согласно которому программное средство (ПС, программа, библиотека, веб-приложение и др.) разделяется на отдельные обособленные единицы, называемыми модулями. Модульный подход обычно является средством упрощения задачи проектирования ПС и распределения процесса разработки ПС между группами разработчиков. При разбивке ПС на модули для каждого модуля описывается его функциональность и назначение, а также связь с другими частями программы (модулями).

Модульное программирование — разработка и структуризация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определенным правилам. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок. Аппаратно-зависимые подзадачи могут быть отделены от других подзадач, что улучшает мобильность создаваемых программ и дает возможность повторного использования готовых модулей в других проектах [2].

Основная часть

При использовании модульного программирования при разработке программы можно выделить плюсы и минусы данного подхода [3]:

Плюсы:

- возможность разрабатывать модули с использованием различных языков программирования;
- модуль является естественной единицей локализации имен;
- локализация места ошибки: обычно исправление ошибки внутри одного модуля не влечет за собой исправление в других модулях (естественно, если это свойство будет выполняться только при хорошей разбивке программы на части, с малым числом связей между модулями);
- возможность повторного использования разработанных модулей в других программах.

Минусы:

- модули не являются совсем уж независимыми друг от друга: между ними существуют связи, то есть один модуль иногда может завязываться на переменные, константы и программный код из другого модуля;
- перед тем, как будет произведен первый запуск программы, необходимо собрать все нужные модули для программы в особую сборку. Этот процесс достаточно сложен, так как кроме собственно объединения всех модулей в одну программу, необходимо проконтролировать и установить все связи между этими частями;
- программа-компилятор, с помощью которой делается сборка программы, не видит сразу всех модулей и не может проконтролировать, установлены ли правильно связи между модулями.

Несмотря на выделенные недостатки, преимущества модульного программирования настолько огромны, что сейчас оно является основным способом разработки сложного программного обеспечения.

Для примера, хотелось бы рассмотреть модульный подход при разработке АСУФП (автоматизированной системы управления фармацевтическим предприятием). Сначала весь проект был разделен на 3 очереди и задокументирован порядок разработки отдельных подсистем в техническом задании. В данном примере очереди представляют собой подразделения предприятия и схожи с его инфраструктурой, т.е. на данном этапе, модулем является отдел:

- Отдел закупок;
- Отдел продаж;
- Отдел складских помещений;
- Отдел цехов и т.д.

Примерная схема работы предприятия, из которой становится интуитивно понятно, на какие модули можно разделить проект, представлена на рисунке.

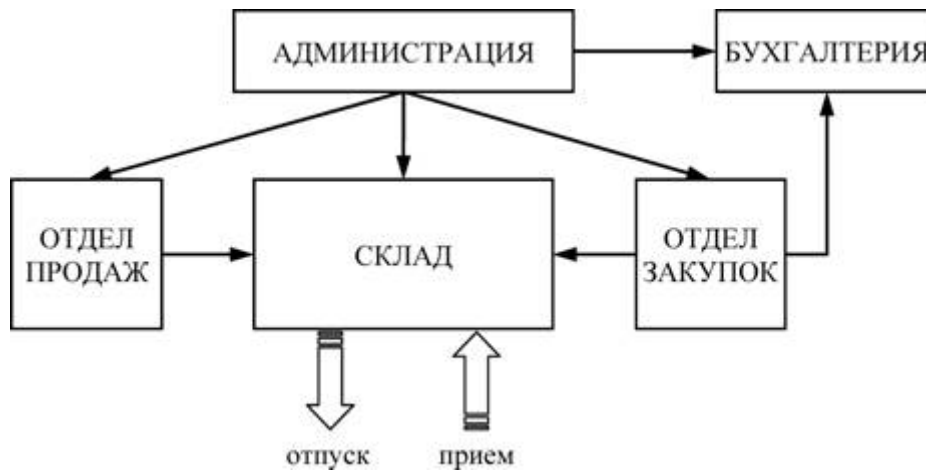


Рисунок – Примерная схема работы предприятия

При разработке системы «Управление закупками» был создан модуль для отдела закупок, который был разделен на более мелкие задачи:

- регистрация сырья и материалов;
- создание и ведение норм;
- планирование закупок;
- регистрация и ведение номенклатуры сырья и материалов;
- ведение платежных поручения и требований и т.д.

Каждый модуль, можно разделить на ещё более мелкие задачи, распределив их между группами разработчиков, что увеличит скорость разработки системы, снизит время и силы, затрачиваемые на тестировании и поиск ошибок в программе.

В данной работе было рассмотрено:

- что такое модульное программирование;
- назначение;
- понятия модуля и модульности;
- основные плюсы и минусы;
- разработка программного продукта с использованием модульного подхода.

Выводы

При разработке сложного программного обеспечения основным способом разработки является модульное программирование, т.к. имеет ряд преимуществ.

Функциональная постановка и описание исходной задачи – разбивка первоначальной большой задачи на ряд более мелких, функционально самостоятельных подзадач – модулей. Для отлаженной и бесперебойной работы программного обеспечения модули желательно делать обособленными. Допускается реализовывать связь между модулями только по входным и выходным данным. Каждый модуль в функциональной декомпозиции представляет собой «черный ящик» с одним входом и одним выходом. Использование модульного подхода дает возможность безболезненно производить модернизацию программы в процессе ее тестирования, эксплуатации и облегчает ее сопровождение. Дополнительно модульный подход позволяет разрабатывать части программ одного проекта с использованием разных языков программирования, после чего с помощью компоновочных средств объединять их в единую сборку.

Реализуемые решения должны быть простыми и ясными. Если назначение модуля непонятно, то это говорит о том, что разбивка начальной или промежуточной задачи

на более простые была проведена недостаточно качественно. В этом случае необходимо повторно проанализировать задачу и, возможно, провести дополнительное деление на подзадачи.

Реализуемые задачи можно разделить между командой программистов, что ускоряет разработку программы.

С использованием вышеизложенного материала была разработана система «Управление закупками», которая включает множество модулей и сама является модулем в ещё более крупной системе.

Литература / References

1. Разработка структуры программы и модульное программирование [Электронный ресурс] – Режим доступа : http://www.vit-prog.narod.ru/page/TRPP/section_1/subject_1.3.html. – Дата доступа: 02.11.2013.

Razrabotka struktury programmy i modulnoye programmirovaniye [Electronic resource] Mode of access: http://www.vit-prog.narod.ru/page/TRPP/section_1/subject_1.3.html. – Date of access: 02.11.2013.

2. Принцип модульного программирования // Большая энциклопедия нефти газа [Электронный ресурс] – Copyright 2008-2013 by «Знание» – Режим доступа : <http://www.ngpedia.ru/id319297p1.html>. – Дата доступа: 05.11.2013.

Printsip modulnogo programmirovaniya // Bolshaya entsiklopediya nefi gaza [Electronic resource] – Copyright 2008-2013 by «Znaniye» – Mode of access: <http://www.ngpedia.ru/id319297p1.html>. – Date of access: 05.11.2013.

3. Модульное программирование [Электронный ресурс] – Режим доступа: http://arch.cs.msu.su/Text/Chapter_09. – Дата доступа: 01.11.2013.

Modulnoye programmirovaniye [Electronic resource] – Mode of access: http://arch.cs.msu.su/Text/Chapter_09. – Date of access: 01.11.2013.