

A.Zh.Amirov, M.K.Baimuldin, Yu.K.Shakirova, G.B.Abildaeva, N.K.Savchenko

Enterprise Resource Planning — system planning resources of enterprise

In article systems oplanning resources of enterprise are considered, is carried out characteristics of ERP system. Aspects ERP system of influence on the world of business and on the world of information technologies are offered. Tasks are offered at Enterprise resource planning design for effective adoption of administrative decisions. Questions in the course of reengineering are also considered. Functional blocks realizing in Enterprise resource planning according to APICS are considered. Examples of problems with ERP systems in introduction and test of changes of system are presented. Modern Enterprise resource planning possess the high-level and integrated environment of development.

References

- 1 O'Leary D. *Enterprise resource planning. Modern planning and enterprise resource management. Choice, introduction, operation*, Moscow: JSC Vershina, 2004, p. 272.
- 2 Piterkin C.B., Oladov N.A., Isaev D.V. *Just in time for Russia. Practice of use of ERP systems*, 2nd prod., Moscow: Alpina Publisher, 2010, p. 368.
- 3 *Management of processes* / Under the editorship of Y. Becker, L. Vilkov, V. Taratukhin, M. Kugeler, M.Rozemann, Moscow: Eksmo, 2008, p. 384.
- 4 Hamilton S. *Management of chains of deliveries with Microsoft Axapta*, Moscow: Alpina Publisher, 2005, p. 349.

УДК 378:658.336.3

Ю.Н.Антипов¹, Г.Т.Омаров², Б.К.Шаяхметова³

¹Калининградский технический университет, Россия;

²Карагандинский экономический университет Казпотребсоюза;

³Карагандинский государственный университет им. Е.А.Букетова (E-mail: kazahzavod@mail.ru)

Методика преподавания структурного программирования

В статье исследовано структурное программирование — одна из парадигм современной информатики, даны рекомендации по методике преподавания этого метода. Отмечено, что структурный подход является интенсивно развивающимся направлением теоретического и прикладного программирования. Его применение позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, модифицируемые программные системы. Этим объясняется интерес к структурным языкам программирования. С этой точки зрения и исследована методика преподавания структурного программирования.

Ключевые слова: структурное программирование, методика преподавания, языки программирования, программные системы, современные технологии, цикл, ветвление, подпрограмма, декомпозиция, процедуры.

Современные потребности образовательных, производственных и коммерческих структур выдвинули программное обеспечение наиболее важных видов деятельности. И с этой точки зрения необходимо изучать различные подходы к составлению программ. Анализ большинства подходов, изучаемых в высшей школе, показал, что наиболее используемым является структурный подход к программированию, который возник в 60–70-е годы XX в. Однако в учебных программах ему уделяется, как нам кажется, неоправданно мало времени — порядка 8–12 ч. Мы хотели бы восполнить этот пробел и в связи с этим изложить методику преподавания названного выше подхода. Создание программ с использованием новых инновационных методов требует смены технической оснащённости методической и содержательной базы учебного процесса. Текущие проблемы компьютерного моделирования приводят к созданию подходов к программированию, более полно отражающих природу моделируемых процессов. Наиболее полно отвечает решению указанных проблем структурный подход к

разработке программных средств, которые способствуют переходу к новой парадигме программирования. Современные научные разработки в области педагогической информатики продиктованы стремлением приблизить содержание преподавания программирования к передовому состоянию науки, найти пути более эффективного внедрения новых информационных технологий в учебный процесс [1].

Структурный подход является интенсивно развивающимся направлением теоретического и прикладного программирования. Его применение позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, модифицированные программные системы. Этим объясняется интерес к структурным языкам программирования. Таким образом, структурный подход к программированию представляет собой совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения.

Здесь же преподаватели, ведущие занятия по информационным дисциплинам, должны будут привести стандартные определения, например: структура программы — искусственно выделенные программистом взаимодействующие части программы. Использование рациональной структуры устраняет проблему сложности разработки; делает программу понятной людям; повышает надежность работы программы при сокращении срока её тестирования и сроков разработки вообще.

Методики, используемые преподавателями и студентами, зависят от комплекса различных факторов, как, например, профессионализм, уровень подготовки студента, техническая обеспеченность и многих других, и, конечно же, уровня информационной подготовленности. Информационная подготовленность — это понимание не только пакета элементарных компьютерных программ, а осознание значимости и огромного потенциала процессов информатизации, познание правовых, социальных, психологических и других аспектов функционирования и практического использования совокупности специальных программ, умения максимально эффективно применить имеющиеся возможности и найти новые пути к преодолению существующих проблем.

Возвращаясь к структурному программированию, необходимо отметить еще одну существенную сторону, которую преподавателям нужно вложить студентам.

Итак, часто некоторую последовательность инструкций требуется повторить в нескольких местах программы. Чтобы программисту не приходилось тратить время и усилия на копирование этих инструкций, в большинстве языков программирования предусматриваются средства для организации. Таким образом, программист получает возможность присвоить последовательности инструкций произвольное имя и использовать это имя в качестве сокращенной записи в тех местах, где встречается соответствующая последовательность инструкций. Подпрограмма — некоторая последовательность инструкций, которая может вызываться в нескольких местах программы. Преподавателю необходимо, чтобы студенты четко знали, что программа представляет собой структуру, построенную из трех типов базовых конструкций:

- последовательное исполнение — однократное выполнение операций в том порядке, в котором они записаны в тексте программы;
- ветвление — однократное выполнение одной из двух или более операций, в зависимости от выполнения некоторого заданного условия;
- цикл — многократное исполнение одной и той же операции до тех пор, пока выполняется некоторое заданное условие (условие продолжения цикла).

В программе базовые конструкции могут быть вложены друг в друга произвольным образом, но никаких других средств управления последовательностью выполнения операций не предусматривается.

Повторяющиеся фрагменты программы (либо не повторяющиеся, но представляющие собой логически целостные вычислительные блоки) могут оформляться в виде т.н. подпрограмм (процедур или функций). В этом случае в тексте основной программы, вместо помещенного в подпрограмму фрагмента, вставляется инструкция вызова подпрограммы. При выполнении такой инструкции выполняется вызванная подпрограмма, а после чего исполнение программы продолжается с инструкции, следующей за командой вызова подпрограммы.

Педагог должен быть уверен, что студенты верно поняли структурное программирование и последовательность составления программ.

Разработка программы ведётся пошагово, методом «сверху вниз». Сначала пишется текст основной программы, в котором вместо каждого связанного логического фрагмента текста вставляется вызов подпрограммы, которая будет выполнять этот фрагмент. Вместо настоящих, работающих подпрограмм в программу вставляются «заглушки», которые ничего не делают. Полученная программа про-

веряется и отлаживается. После того, как программист убедится, что подпрограммы вызываются в правильной последовательности (т.е. общая структура программы верна), подпрограммы-заглушки последовательно заменяются на реально работающие, причём разработка каждой подпрограммы ведётся тем же методом, что и основной программы. Разработка заканчивается тогда, когда не останется ни одной «затычки», которая не была бы удалена. Такая последовательность гарантирует, что на каждом этапе разработки программист одновременно имеет дело с обзримым и понятным ему множеством фрагментов и может быть уверен, что общая структура всех более высоких уровней программы верна. При сопровождении и внесении изменений в программу выясняется, в какие именно процедуры нужно внести изменения, и они вносятся, не затрагивая части программы, непосредственно не связанные с ними. Это позволяет гарантировать, что при внесении изменений и исправлении ошибок не выйдет из строя какая-то часть программы, находящаяся в данный момент вне зоны внимания программиста.

Педагог должен объяснить студентам, обучающимся на факультете по информационным специальностям, что в основе структурного программирования лежит представление программы в виде иерархической структуры блоков, т.е. лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40–50 операторов) подпрограмм. Педагогам необходимо отметить, что, вероятно, наиболее общая тактика программирования состоит в разложении процесса на отдельные действия: функционального описания на подфункции, а соответствующих программ — на отдельные инструкции. На каждом таком шаге декомпозиции нужно удостовериться, что решения частных задач приводят к решению общей задачи; выбранная последовательность отдельных действий разумна; выбранная декомпозиция позволяет получить инструкции, в каком-либо смысле более близкие к языку, на котором будет реализована программа [2].

Опыт показывает, что программы, составленные с помощью структурного программирования, в значительной степени способствуют формированию и развитию интеллектуального потенциала обучаемого совершенствованию форм и содержания учебного процесса, внедрению инновационных методов в обучении и дают возможность разрешать на новом уровне имеющиеся проблемы.

Крупные функции разбиваются на подфункции до достижения подфункции размера модуля — подпрограммы (процедуры или функции) языка программирования, к которым предъявляются особые дополнительные требования. Итак, модуль — фундаментальное понятие и функциональный элемент технологии структурного программирования, или модуль — это подпрограмма, но оформленная в соответствии с особыми правилами. Существует несколько правил, которым должны удовлетворять модули: например, модуль должен иметь один вход и один выход и выполнять строго однозначную функцию, которая описывается простым распространённым предложением естественного (русского) языка или даже предложением без сказуемого, или модуль должен обеспечивать компиляцию, независимую от других модулей, с «забыванием» всех внутренних обозначений модулей. Существуют и другие правила описания модулей.

В понятие структуры программы включаются состав и описание связей всех модулей, которые реализуют самостоятельные функции программы и описание носителей данных, участвующих в обмене как между отдельными подпрограммами, так и вводимые и выводимые с/на внешних устройств. Как известно, сердцевиной современных информационных технологий в сфере образования являются педагогические программы, средства, электронные дидактические пособия, учебно-методические комплексы, обучающие тренажеры, использование которых позволяет преподавателю эффективно спроектировать педагогический процесс и совершенствовать его в дальнейшем. Качественное оформление и удобный интерфейс, оживлённый анимацией, делают процесс обучения более привлекательным, менее рутинным. Использование гипертекста, графики, анимации, аудио- и видеoinформации в содержании позволяет увеличить эффективность и качество преподавания, а наличие различных тестов и практических упражнений повышает объективность и качество контроля знаний. Именно с этой точки зрения необходимо изучать методику преподавания структурного программирования в высшей школе.

Теперь рассмотрим написанный выше материал в разрезе создания и апробирования структурного программирования. Содержание курса, на основе которого изучаются этапы создания программных продуктов для сложных систем, — это системный анализ, который задает общие, верные для всей системы в целом ограничения. Необходимость системного анализа явно проявляется, когда формируется интерфейс программного обеспечения с другими элементами (аппаратурой, персоналом, базами данных).

Содержание данного курса, с помощью которого осуществляется подготовка студентов информационных специальностей к профессиональной деятельности, определено современным уровнем развития программного обеспечения. Для обоснования содержания курса использовался метод экспертных групповых оценок. В качестве экспертов были задействованы квалифицированные специалисты по программному обеспечению, работающие во внедренческих фирмах, и преподаватели вузов. Обработка результатов проводилась с помощью автоматизированной информационной системы анализа логической структуры учебного материала. Проведенное исследование позволило скорректировать и предположить программу курса, отвечающего требованиям, предъявляемым к современному специалисту в области программного обеспечения.

В случае сложной, большой программы необходимо овладеть специальными приемами получения рациональной структуры программы. И здесь преподаватели, ведущие занятия по структурному программированию, должны подчеркнуть, что рациональная структура программы обеспечивает почти двукратное сокращение объема программирования и многократное сокращение объемов и сроков тестирования, а следовательно, принципиально снижает затраты на разработку.

Подчиненность модулей удобно изображать схемой иерархии. Схема иерархии отражает только подчиненность подпрограмм, но не порядок их вызова или функционирование программы. Схема иерархии обычно дополняется расшифровкой функций, выполняемых модулями. Необходимо также акцентировать внимание студентов на том, что до составления схемы иерархии целесообразно составить внешние спецификации программы и составить функциональные описания программы вместе с описанием переменных — носителей данных. Особое внимание надо уделить иерархии типов структурированных данных и их комментированию. Декомпозиция программы на подпрограммы производится по принципу от общего к частному, более детальному. Вообще процесс составления функционального описания и составления схемы иерархии является итерационным. Выбор наилучшего варианта является многокритериальным.

Расчленение должно обеспечивать удобный порядок ввода частей в эксплуатацию.

Возвращаясь к структурному программированию, заметим, что реализация программы (кодирование, сборка, тестирование) должна вестись по разработанному заранее плану и начинаться с верхних модулей схемы иерархии. Недостающие модули нижних уровней заменяются «заглушками», которые представляют собой простейшие подпрограммы: либо без действий; либо выводящие в файл отладки входные данные; либо возвращающие в стоящие выше модули тестовые данные (которые обычно присваиваются внутри «заглушки»); либо содержащие комбинацию этих действий.

Анализ показывает, что такой подход ведет к формированию у студента концептуального мышления будущего специалиста, способного быстро адаптироваться в динамичной изменчивости современного общества, ибо всё чаще работодателю требуется профессионал, готовый к приобретению и способный к быстрой переориентации.

В этом смысле структурный подход к программированию представляет собой как методологию, так и технологию создания программ. В свое время его внедрение обеспечило повышение производительности труда программистов при написании и отладке программ; получение программ, которые состоят из модулей и пригодны для сопровождения; создание программ коллективом разработчиков; окончание создания программы в заданный срок. Надо заметить, что всплеск компьютерных и телекоммуникационных технологий в образовании привел к формированию нового студенчества. Это студенчество воспитано на основе тестирования качества знания и других методов измерения уровня знаний, умений и навыков, что порождает комплекс информационных процессов, в автоматизации которых в вузах используются компьютерная техника и специализированное программное обеспечение. И с этой точки зрения структурный подход к программированию воспринял и использует многие методы из области проектирования сложных технических систем. Среди них блочно-иерархический подход к проектированию сложных систем, стадийность создания программ, нисходящее проектирование, методы оценки и планирования.

Структурный подход рекомендует соблюдать следующие принципы при создании программного изделия [3]:

- модульность программ;
- структурное кодирование модулей программ;
- нисходящее проектирование рациональной иерархии модулей программ;
- нисходящая реализация программы с использованием заглушек;
- осуществление планирования на всех стадиях проекта;
- сквозной структурный контроль программных комплексов в целом и составляющих их модулей.

Методики, используемые преподавателями, зависят от комплекса различных факторов: профессионализма, уровня подготовки студента, технической обеспеченности и многих других, и, конечно, от уровня информационной подготовленности. Поэтому модульность программ характеризуется тем, что вся программа состоит из модулей. Некоторые смысловые группы модулей сосредотачиваются в отдельных файлах. Например, в отдельных файлах (*Unit*) могут быть сосредоточены модули текстового редактора и модули иерархического меню.

Структурное кодирование модулей программ заключается в особом оформлении их текстов. У модуля должен быть легко различимый заголовок с комментарием, поясняющим функциональное назначение модуля. Имена переменных должны быть мнемоническими. Суть переменных и порядок размещения в них информации должны быть пояснены комментариями, а код модуля закодирован с использованием типовых алгоритмических структур и использованием отступов.

Нисходящее проектирование рациональной иерархии модулей программ заключается в выделении первоначально моделей самого верхнего уровня иерархии, а затем подчиненных модулей.

Нисходящая реализация программы состоит в первичной реализации группы модулей верхних уровней, которые называются ядром программы, и далее постепенно, в соответствии с планом, реализуются модули нижних уровней. Необходимые для линковки программы, недостающие модули имитируются «заглушками».

Осуществление планирования на всех стадиях проекта позволяет первоначально спланировать как состав стадий, так и продолжительность всех этапов работ. Такое планирование позволяет завершить разработку в заданный срок при заданных затратах на разработку. Далее планируются порядок и время интеграции модулей во все расширяющееся ядро. Намечаются мероприятия по тестированию программы от ранних до заключительных этапов.

Сквозной структурный контроль заключается в соблюдении ранее намеченного плана тестирования, который охватывает период от разработки внешних спецификаций, далее внутренних спецификаций и их корректировки в период реализации вплоть до приемо-сдаточных испытаний. Составляющие программу модули тестируются как во время написания их кода, так и при автономном тестировании, инспекции их исходного кода, при тестировании сразу по подключении к ядру.

Список литературы

- 1 Шаяхметова Б.К., Омаров Т.Е. О предполагаемых подходах к совершенствованию содержания образования специалистов по информационным системам // Вестн. Караганд. ун-та. Сер. «Педагогика». — 2006, № 1 (41). — С. 92–95.
- 2 Йодан Э. Структурное проектирование и конструирование программ / Э. Йодан; пер. с англ. В.В. Фролова, О.А. Темлицкого / Под ред. Л.Н. Королева. — М.: Мир, 1979. — 360 с.
- 3 Хьюз Дж., Мичтом Дж. Структурный подход к программированию / Хьюз Дж., Мичтом Дж.; пер. с англ. Э.М. Куиру и А.Л. Александрова / Под ред. В.Ш. Кауфмана. — М.: Мир, 1980. — 278 с.

Ю.Н.Антипов, Г.Т.Омаров, Б.К.Шаяхметова

Құрылымдық бағдарламалауды оқыту әдістемесі

Мақалада қазіргі информатика — құрылымдық бағдарламалау парадигмасының бірі зерттелді. Бұл әдіс сабақ берудің әдістемесі бойымен кепілдемені береді. Құрылымдық көзқарас теоретикалық және қолданбалы бағдарламаның жіті дамуының бағыты болатыны айтылды. Оның қолдануы жақсы құрылымды, сенімді пайдалануды, өңделген бағдарлама жүйелерді жетілдіруіне мүмкіндік береді. Осымен құрылымдық бағдарламалау тілдеріне назар аударуын түсіндіреді. Бұл көзқарас арқылы құрылымдық бағдарламаның оқыту әдістемесі зерттелді.

Yu.N.Antipov, G.T.Omarov, B.K.Shayakhmetova

The method of structured programming teaching

In the work structured programming was research, one of the paradigms of modern informatics, also were provided some recommendations of teaching this method. States that a structured approach is intensively developing area of theoretical and applied programming. Its application allows to develop a well-structured, reliable in operation, the modified software systems. Therefore, it explains the interest in the structural programming languages. From this viewpoint, methods of teaching structured programming are being explored.

References

- 1 Shayakhmetova B.K., Omarov T.E. Bull. of KarSU. Ser. «Pedagogy», 2006, 1 (41), p. 92–95.
- 2 Yodan E. *Structural design and construction programs* / E. Yodan; trans. from eng. V.V.Frolova, O.A.Temlitskogo: ed. L.N. Koroleva, Moscow: Mir, 1979, 360 p.
- 3 Kh'yuz Dzh., Michtom Dzh. *Structured approach to programming* / Kh'yuz Dzh., Michtom Dzh. trans. from English. E.M. Kuiru and A.L. Alexandrov, ed. V.Sh. Kaufman, Moscow: Mir, 1980, 278 p.

UDC 539.3 +534.1

E.Arinov¹, N.A.Ispulov²

¹*O.A.Baykonurov Zhezkazgan University;*

²*S.Toraygyrov Pavlodar State University (E-mail: arinov91@mail.ru)*

The voltage concentration in the vicinity of spherical mine workings of deep foundations

A solution for the determination of voltage concentrations in the vicinity of spherical mine workings of deep foundations is given. A system of differential equations of equilibrium in a spherical coordinate system is used for solving the task. The case of spheroidal deformation in axisymmetric loading is considered.

Key words: mining, elastic deformation, movement vectors, differential equations, boundary value problems.

The system of differential equations of equilibrium in a spherical coordinate system has the following form [1]:

$$\begin{aligned} \frac{\partial \sigma_{11}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{12}}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial \sigma_{13}}{\partial \lambda} + \frac{1}{r} (2\sigma_{11} - \sigma_{22} - \sigma_{33} + \sigma_{12} \operatorname{ctg} \theta) &= 0, \\ \frac{\partial \sigma_{12}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{22}}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial \sigma_{23}}{\partial \lambda} + \frac{1}{r} [(\sigma_{22} - \sigma_{33}) \operatorname{ctg} \theta + 3\sigma_{12}] &= 0, \\ \frac{\partial \sigma_{13}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{23}}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial \sigma_{33}}{\partial \lambda} + \frac{1}{r} (3\sigma_{13} + 2\sigma_{23} \operatorname{ctg} \theta) &= 0, \end{aligned} \tag{1}$$

where σ_{ij} ($i, j = 1, 2, 3$) are the components of voltage tensor; r, θ, λ are the spherical coordinates, and r is the radius, θ is the co latitude, λ is the longitude.

Since gasholder and oil storage are arranged usually among rocks of incompressible material, we use appropriate physics' law that establishes the connection between the components of voltage and strain tensors:

$$\sigma_{ij} - \delta_{ij} \sigma = 2G \varepsilon_{ij}, \tag{2}$$

where δ_{ij} is the Kronecker sign, σ is the average voltage; ε_{ij} is the components of strain tensor; G is the shear modulus.

The components of the strain tensor and elastic movements are connected by Cauchy correlations in the spherical coordinate system:

$$\begin{aligned} \varepsilon_{11} &= \frac{\partial u_1}{\partial r}, & \varepsilon_{22} &= \frac{1}{r} \frac{\partial u_2}{\partial \theta} + \frac{u_1}{r}; \\ \varepsilon_{33} &= \frac{1}{r \sin \theta} \frac{\partial u_3}{\partial \lambda} + \frac{u_2}{r} \operatorname{ctg} \theta + \frac{u_1}{r}; \\ \varepsilon_{12} &= \frac{1}{2} \left(\frac{\partial u_2}{\partial r} + \frac{1}{r} \frac{\partial u_1}{\partial \theta} - \frac{u_2}{r} \right); \end{aligned}$$