

ТЕОРИЯ И МЕТОДИКА ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ (13.00.08)

УДК 004.432.2:519.682.2

DOI: 10.24160/1993-6982-2019-3-142-147

Структурное программирование как средство совершенствования компьютерной подготовки бакалавров

Л.В. Чуркина, Н.В. Чибизова, А.А. Горкина

Рассмотрены проблемы компьютерной подготовки инженеров в вузах, возникающие в результате недостаточного понимания основной цели подготовки. Если ее рассматривать как часть общей математической подготовки и определить цель, как обучение студента решению задач на компьютере, то эта цель достигается путём обучения структурному программированию, представляющему собой современный способ решения задач на компьютере.

К настоящему моменту программирование концептуально полностью сформировалось как наука, что принципиально меняет подход к обучению программированию. Идеология структурного программирования — основа для понимания процесса программирования как формализованного процесса решения задач.

Структурное программирование опирается на базовый язык Паскаль, который был создан для обучения процессу программирования. В нем ярко отражены возможности компьютера как инструментария, они необходимы и достаточны для решения задачи любой сложности, поскольку в Паскале строго определён набор типов данных, обеспечивающий процесс решения задач.

Обучение структурному программированию — это обучение технологии решения задач на компьютере, которая является основой компьютерной грамотности любого выпускника вуза. В основу технологии положен нисходящий способ разработки программы, включающей правила композиции и декомпозиции подзадач.

Изучая программирование как технологический процесс решения задач, студент уже на начальном этапе своего обучения должен приобретать навыки решения задач на компьютере и развивать алгоритмическое мышление. Подобный опыт полезен будущему инженеру, специалисту по технологиям в своей профессии.

Ключевые слова: задача, структурное программирование, нисходящее проектирование, методология программирования.

Для цитирования: Чуркина Л.В., Чибизова Н.В., Горкина А.А. Структурное программирование как средство совершенствования компьютерной подготовки бакалавров // Вестник МЭИ. 2019. № 3. С. 142—147. DOI: 10.24160/1993-6982-2019-3-142-147.

Structured Programming as a Means for Improving the Computer Training of Bachelors

L. V. Churkina, N. V. Chibizova, A. A. Gorkina

The article addresses the problems associated with computer training of engineers at higher schools, which are stemming from insufficiently clear understanding of the main training goal. If this training is seen as a part of the overall mathematical education with defining its goal as educating a student to solve problems on a computer, this goal is achieved by teaching to structured programming, which is a modern method of solving problems on a computer.

Conceptually computer programming has presently been fully shaped as a science, a circumstance that fundamentally changes the very approach to education for programming. The structured programming philosophy forms the basis for understanding the programming process as a formalized problem solution process.

Structured programming is based on the basic programming language PASCAL which was developed for teaching to computer programming. This language clearly reflects the capabilities of a computer as a tool; they are necessary and sufficient for solving problems of any complexity because the PASCAL language strictly defines the set of data types supporting the problem solving process.

Education for structured programming is learning the technology of solving problems on a computer which is the basis of computer literacy of any higher school graduate. The top-down program development method, which includes the rules for aggregation and decomposition of sub-problems, has been laid at the heart of this technology.

In learning computer programming as a technological process for solving problems, a student should gain skills of solving problems on a computer and develop an algorithmic way of thinking already at the initial stage of his or her education. This experience will be useful for the future engineer, a specialist on technologies in his or her profession.

Key words: problem, structured programming, top-down designing, computer programming methodology.

For citation: Churkina L.V., Chibizova N.V., Gorkina A.A. Structured Programming as a Means for Improving the Computer Training of Bachelors. Bulletin of MPEI. 2019;3:142—147. (in Russian). DOI: 10.24160/1993-6982-2019-3-142-147.

Введение

Сложность проблем предмета и методов обучения в области компьютерной подготовки будущих инженеров постоянно растёт, отражая быструю смену поколений машин и программного обеспечения.

В настоящее время ситуацию в области базовой компьютерной подготовки можно оценить как кризисную. Подтверждением этому служит противоречивый спектр требований различных кафедр — от использования «модного» языка Питон как базового языка обучения программированию до (согласно мнению «программирование отмирает!») знакомства с текстовыми редакторами и электронными таблицами. Менялись языки программирования, названия базовых курсов, но даже последнее наименование «Информатика», совпадающее со школьным названием, не способствует проведению чётких границ со школьной «Информатикой». В школах отсутствует единая программа по данному предмету, выпускники приходят в вуз, имея разный уровень подготовки по информатике (и, в частности, программированию), некоторые даже нулевой.

К сожалению, в современной литературе теме научного подхода к преподаванию программирования в высшей школе внимания не уделяется. Ведущие журналы не публикуют статей по этой теме, а существующие специализированные издания ориентированы на преподавание информатики в школе.

Цель обучения

Программирование — это способ решения задачи с помощью компьютера. Обучение программированию — это обучение технологии решения задачи на компьютере.

Ершов А.П. ещё в 1960-е гг. писал: «...Законы программирования смыкаются с математическим образованием, образуя единый, но ещё не построенный фундамент воспитания операционного и комбинаторного мышления, способного к абстракции, рассуждению и действию...» [1].

Существует нерасторжимая связь операционного знания и алгоритмического мышления с другими компонентами образования.

В периоды зарождения и формирования программирования как науки не были чётко определён состав базовых понятий и создана теория программирования, поэтому оно было доступно только избранным. С появлением языков программирования высокого уровня стало намного проще писать программы. Программные продукты начали бурно внедряться во все сферы деятельности человека. Это был период расцвета. Однако на рубеже 1970-х гг. в программировании появились трудности, которые привели к кризису: многие проекты сдавали не в срок или вовсе не завершали, а в готовых находили ошибки. Этому способствовал ряд причин. Программирование стало массовым, к нему приобщились и непрофессионалы, поэтому программы должны были легко восприниматься и давать представление о возможностях машины.

При разработке больших программных систем требовались фундаментальные знания по программированию. Производство программ должно было подчиняться определённой технологии программирования, чтобы можно было планировать процесс его производства.

Программы писались хаотично. По тексту программы трудно было понять логику их выполнения. Одной из первых мер по улучшению ситуации с читабельностью программ стало решение ограничить использование в программировании оператора безусловного перехода GOTO. Было даже введено ограничение на его использование — не более 20% от общего количества операторов, но полностью проблему это не решило.

С целью создания ясных, легко понимаемых и отлаживаемых программ и возможностью планирования сроков разработки программных систем была предложена новая дисциплина, названная структурным программированием. В её основу заложен принцип, что программа легко понимаема, если текстуальная упорядоченность программы соответствует порядку вычисления [2]. Другими словами, читая программу, мы должны легко представлять порядок выполняемых ею действий.

Для выполнения данного принципа алгоритм должен быть сконструирован не беспорядочно, а с использованием чётко определённых структурных правил композиции, составляющих базис программирования: следование, выбор, повторение.

Структурированной называется программа, которая строится с использованием только этих трёх правил композиции (управляющих структур).

С этого момента программирование начало развиваться как наука. После того, как, с точки зрения научного познания, предмет и метод были сформированы, продолжала складываться система познания, развивалась методология программирования.

С появлением теории программирования, формализации процесса, развивающейся методологии была решена основная проблема адаптации решаемых задач к «узкому руслу» возможностей компьютера. С созданием методологии структурное программирование стало инвариантным относительно класса решаемых задач, а также той среды на компьютере, в которой они решались. Кризис в программировании был преодолен.

Из вышесказанного следует, что цель компьютерной подготовки — обучить студентов формализованному процессу решения задач на компьютере.

Базовый язык обучения

В начальный период перехода к структурному программированию используемые процедурно-ориентированные языки (Фортран, Бейсик) не содержали главной метафоры: в них плохо отражалась работа компьютера, поэтому они не могли быть использованы как языки обучения структурному программированию. Использовались псевдокоды. В России был разработан специальный псевдокод (русский алгоритмический язык — РАЯ) [3], на котором разрабатывались структурированные программы, а с него делался перевод на языки программирования.

В конце 1960-х гг. был создан язык Паскаль как язык для обучения структурному программированию [4]. Следует отметить, что только на базе структурного программирования можно обучать процессу программирования, других путей пока не существует. Изучение языков не может заменить обучение процессу программирования. Все средства языка Паскаль направлены на этот процесс. Синтаксис содержит строгую иерархию основных понятий программирования. Синтаксические правила компактны и не имеют избыточности и неоднозначности, средства обеспечивают создание структурированных, эффективных и надёжных программ. Состав типов данных, как простых, так и структурных, позволяет решать задачи различных классов, создавать сложные структуры из стандартных типов данных. Набор стандартных типов языка определен как необходимый и достаточный для создания любых типов данных, требуемых в задачах.

Благодаря своей концептуальной строгости Паскаль быстро приобрёл такую популярность среди программистов, как ни один из предшествующих языков. Поскольку в нем ярко выражена главная метафора процедурно-ориентированных языков (ориентированных на специфику работы компьютера), то он и сейчас

является наилучшим языком для обучения программированию [5].

Инженер — это специалист, владеющий определённым инструментарием для решения своих задач, поэтому при изучении программирования ему важно разобраться в базовом инструментарии, т. е. возможностях компьютера, которые отображены в Паскале.

Паскаль оказал большое влияние на развитие традиционных языков, которые модифицировались в «паскалеподобную» форму с учётом структурированного ядра. Переход к другим языкам потребовал только перекодировки с учётом специфики самих языков. Фактически, при обучении программированию на Паскале использовался надязыковый подход к программированию, не зависящий от конкретного языка программирования.

С каждым годом количество языков, их версий, программных систем увеличивается. Но любое совершенствование языка, как правило, приводит к добавлению новых функций в сторону его проблемной ориентации, которые отдаляют пользователя от компьютера. При этом теряется полнота связей, что снижает качество компьютерного образования.

Требования некоторых кафедр к выбору своего кафедрального языка в качестве базового при обучении программированию способствуют воспитанию квалифицированного потребителя. Если языки не удовлетворяют требованиям базового языка, то много дополнительного времени занимает изучение специфики языков. В результате при обучении у студента не формируются базовые фундаментальные представления и необходимые практические навыки.

Так, нельзя использовать язык Питон (работающий со ссылочными типами данных) для обучения, если студент не понимает отличия обычной переменной от ссылки или динамического массива от статического. С другой стороны, невозможно учить строгому программированию, используя язык Си, допускающий «суммирование» переменных несовместимых типов: целого и логического.

Технология программирования

Под технологией программирования понимается совокупность методов, правил и порядка их применения, обеспечивающих процесс получения в срок качественных программ. Она отражает процесс решения задач на компьютере [6, 7].

На первом этапе создаётся точное и полное описание задачи, называемое спецификацией.

На этапе проектирования при выборе способа принимается во внимание следующее обстоятельство. Структурированность программы является дополнительным бременем при программировании, если не использовать нисходящий способ разработки программы (проектирование сверху-вниз). На этапе становления эта стратегия использовалась интуитивно, являясь за-

логом успешной работы. В ней заключена идея постепенного раскрытия деталей проектируемой программы по мере движения от общей цели, сформулированной на самом высоком уровне в условии задачи, к уровню объектов, выраженных в терминах, «понятных машине».

Идея нисходящей разработки состоит в декомпозиции задачи на точно определённые подзадачи и доказательстве правильности этой декомпозиции. Данная идея положена в основу нисходящего метода разработки программ, который состоит из следующих основных шагов [8].

1. Общая задача разделяется на более простые подзадачи. Определяются связи между задачей и подзадачами (интерфейс) и устанавливается их правильность.

2. Для каждой из подзадач повторяется процесс разделения (декомпозиции) и подтверждается правильность этого разделения.

3. Процесс повторяется до достижения задач настолько простых, что их решение может быть выражено в нескольких строках языка.

Процесс проектирования иерархичен и отображается в виде структурной схемы алгоритма: top-down графа, отражающего схему подчинения задач. Следует подчеркнуть, что при нисходящем проектировании в начале проектируется интерфейс подзадачи (входные и выходные данные), а затем логика решения.

Современная методология программирования позволяет использовать язык Паскаль не только как базовый язык программирования, но и для проектирования. Комментарии используются для неформального описания действий. Рельефная запись структурированных программ, в которых в виде отступов отображается вложенность структурных операторов, делает текст программы наглядным с хорошо отображенной структурой алгоритмов.

Следует отметить, что блок-схемы не могут быть использованы в современной методологии программирования как язык нисходящего проектирования, так как они обладают рядом ограничений:

- в них не отражён полностью базис программирования, отсутствует управляющая структура повторения;
- нет средств для описания данных (интерфейса задачи), что должно быть первоочередным при декомпозиции алгоритма;
- двумерное изображение логического блока не гарантирует отсутствие оператора GOTO в программе.

Присутствующие в блок-схемах два графических блока — функциональный (арифметический) и логический со всякими дорисовками позволяют обходить ограничения, но при этом блок-схемы теряют свой первоначальный универсальный смысл, поэтому нисходящее проектирование проводится на Паскале [8].

Логика программ разрабатывается для достаточно простых подзадач, когда их решение может быть выражено в нескольких строках языка программирования. Логика небольших фрагментов (в пределах одной строки текста) может быть легко описана на Паскале с

использованием трёх правил композиции. Поскольку в структурированной программе чётко отражена логика программы, то фактически Паскаль-программа является собственной блок-схемой.

Основное понятие современных компьютеров — действие участвует при разработке программы в виде абстрактных действий языка (операторы и операции), которые описывают выделяемые подзадачи. Первоначально абстрактные действия записываются как комментарии языка, а затем раскрываются в виде композиционных или декомпозиционных структур. Таким образом, при программировании обучаемый уже на начальном этапе обучения может использовать возможности компьютера [9].

Подобная технология решения задач, в которой применяется нисходящий способ разработки программы, даёт представление об алгоритмическом способе решения. Изучая учебные задачи различного класса, студент приобретает опыт универсального, алгоритмического подхода к решению задач на компьютере.

Процесс обучения

При обучении программированию и выработке алгоритмического способа мышления необходимо научиться решать задачи со структурными типами данных, поскольку программа — это алгоритм плюс данные. Структуры алгоритма и данных тесно взаимосвязаны, структура программы зависит от структуры данных. В Паскаль включено описание стандартных типов данных, из которых возможно построение любых сложных структур данных.

Из таблицы Вирта следует, что существуют близкие аналоги между методами структурирования алгоритмов и структурирования данных.

В Паскаль заложены средства для написания эффективных программ с использованием соответствия структур алгоритмов структурам данных.

Таблица Вирта: связь структур алгоритма и данных

Схема строения	Оператор	Тип данных
Атомарный элемент	Присваивание	Скалярный
Следование	Составной оператор	Запись
Известное число повторений	Оператор цикла с параметром	Массив
Выбор	Условный оператор	Запись с вариантами
Неизвестное число повторений	Итерационный оператор цикла	Последовательность или файл
Рекурсия	Оператор процедуры	Рекурсивный
Универсальный граф	Оператор безусловного перехода	Структура, связанная ссылками

Схема обучения

Исходя из теории структурного программирования и базового языка Паскаль проводится обучение технологии решения задач на компьютере, в каждой из которых заложена методическая нагрузка. Набор задач подбирается преподавателем таким образом, чтобы покрыть объем получаемых студентами знаний по программированию. При подобном системном подходе и правильной расстановке акцентов на различных проблемах обучения возможны решение довольно сложных задач различного класса [10], использование готовых программ и оценка качества применяемых алгоритмов [7, 8].

Благодаря использованию универсального подхода к решению задач процесс перекодирования алгоритма с Паскаля на другие процедурно-ориентированные языки программирования не представляет трудности.

Особое место в процессе обучения занимает разработка качественных программ. Ее основу составляет предупреждение ошибок, начиная с постановки и спецификации задачи. Проблеме тестирования уделяется особое внимание. Разработка тестов, основы для отладки, начинается уже на этапе спецификации. Весь процесс разработки сопровождается документированием.

Литература

1. **Ершов А.П.** Программирование — вторая грамотность [Электрон ресурс] http://ershov.iis.nsk.su/ru/second_literacy/article (дата обращения 08.04.2017).
2. **Дейкстра Э.** Заметки по структурному программированию. М.: Мир, 1972.
3. **Ершов А.П.** Алгоритмический язык в школьном курсе основ информатики и вычислительной техники // Макропроцессорные средства и системы. 1985. № 2. С. 48—51.
4. **Архангельский А.Я.** Язык Паскаль и основы программирования в Delphi. М.: Бином-Пресс, 2008.
5. **Горкина А.А., Чуркина Л.В.** Возможность реализации доступа к классическим системам на современных компьютерах на примере Turbo Паскаль. М.: МЭИ, 2011.
6. **Карпов Ю.Г.** Теория и технология программирования. Основы построения трансляторов. СПб.: БХВ-Петербург, 2012.
7. **Чуркина Л.В., Перевезенцева Е.С., Котарова И.Н.** Технология разработки структурированных алгоритмов. М.: МЭИ, 1988.
8. **Технология** разработки многомодульных программ на языке Паскаль. М.: МЭИ, 1998.
9. **Чуркина Л.В.** Структура алгоритмов. Простые переменные. М.: МЭИ, 2011.
10. **Чуркина Л.В.** Решение задач в системе Паскаль с использованием рекурсии. М.: МЭИ, 2011.

Заключение

С использованием структурного программирования при решении задач появилась универсальная методика обучения программированию, направленная на освоение процесса решения задачи на компьютере независимо от класса и конкретного языка программирования. Кроме того, она облегчает перекодировку программ при переходе на другой язык, а также ускоряет процесс освоения нового языка.

Изучая программирование как технологию решения задач, будущий бакалавр приобретает не только навыки программирования, но и опыт по применению технологий, необходимый для своей профессии.

Важность освоения этой дисциплины, развивающей алгоритмическое мышление у студентов, позволяет рассматривать ее как часть математического образования в вузе, так как можно считать, что «образование — это то, что остаётся, когда уже забыто всё, чему учили».

Курс по структурному программированию много лет читается на кафедре прикладной математики НИУ «МЭИ». Он универсален для подготовки студентов разных направлений и профилей. Большой опыт преподавания доказал его высокую эффективность для обеспечения должного уровня компьютерной подготовки при разной степени подготовки студентов в области программирования.

References

1. **Ershov A.P.** Programmirovanie — Vtoraya Gramotnost' [Elektron Resurs] http://ershov.iis.nsk.su/ru/second_literacy/article (Data Obrashcheniya 08.04.2017). (in Russian).
2. **Deykstra E.** Zametki po Strukturnomu Programmirovaniyu. M.: Mir, 1972. (in Russian).
3. **Ershov A.P.** Algoritmicheskiy Yazyk v Shkol'nom Kurse Osnov Informatiki i Vychislitel'noy Tekhniki. Makroprotsessornye Sredstva i Sistemy. 1985;2:48—51. (in Russian).
4. **Arkhangel'skiy A.Ya.** Yazyk Paskal' i Osnovy Programmirovaniya v Delphi. M.: Binom-Press, 2008. (in Russian).
5. **Gorkina A.A., Churkina L.V.** Vozmozhnost' Realizatsii Dostupa k Klassicheskim Sistemam na Sovremennykh Komp'yuterakh na Primere Turbo Paskal'. M.: MEI, 2011. (in Russian).
6. **Karpov Yu.G.** Teoriya i Tekhnologiya Programmirovaniya. Osnovy Postroeniya Translyatorov. SPb.: BKHV-Peterburg, 2012. (in Russian).
7. **Churkina L.V., Perevezentseva E.S., Kotarova I.N.** Tekhnologiya Razrabotki Strukturirovannykh Algoritmov. M.: MEI, 1988. (in Russian).
8. **Tekhnologiya** Razrabotki Mnogomodul'nykh Programm na Yazyke Paskal'. M.: MEI, 1998. (in Russian).
9. **Churkina L.V.** Struktura Algoritmov. Prostyie Peremennyye. M.: MEI, 2011. (in Russian).
10. **Churkina L.V.** Reshenie Zadach v Sisteme Paskal' s Ispol'zovaniem Rekursii. M.: MEI, 2011.

Сведения об авторах:

Чуркина Людмила Витальевна — кандидат технических наук, доцент кафедры прикладной математики НИУ «МЭИ», e-mail: ChurkinaLV@mail.ru

Чибизова Наталья Владимировна — доцент кафедры прикладной математики НИУ «МЭИ», e-mail: ChibizovaNV@mpei.ru

Горкина Анастасия Александровна — ассистент кафедры прикладной математики НИУ «МЭИ», e-mail: GorkinaAA@mpei.ru

Information about authors:

Churkina Ludmila V. — Ph.D. (Techn.), Assistant Professor of Applied Mathematics Dept., NRU MPEI, e-mail: ChurkinaLV@mail.ru

Chibizova Nataliya V. — Assistant Professor of Applied Mathematics Dept., NRU MPEI, e-mail: ChibizovaNV@mpei.ru

Gorkina Anastasiya A. — Assistant of Applied Mathematics Dept., NRU MPEI, e-mail: GorkinaAA@mpei.ru

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов

Conflict of interests: the authors declare no conflict of interest

Статья поступила в редакцию: 09.03.2018

The article received to the editor: 09.03.2018