

Conclusion. The paper discusses how to model a packet network filter that can be used to classify peer-to-peer interactions. Created abstract model obtained during the implementation consists of flexible classification system for network packets and can be used as an auxiliary element for quality of service in the computer networks. Applying the regular expression check operation allows the batch load distribution and parallel execution of the search functions. The result of this was definition and creation of model with optimal complexity, which is mainly applicable to the traffic of the general-purpose computer network and peer-to-peer interaction.

Key words: traffic classification, network filter, peer-to-peer interaction, quality of service in packet networks.

Стаття надійшла 27.04.2017
Прийнято до друку 29.05.2017

УДК 378.147:004.25(07)

PACS 01.50.H-; 01.40.G-; 01.40.Ha; 01.50.H-;
01.50.hv

ГРИШКО Людмила Вениаминовна
Черкаський національний університет
імени Богдана Хмельницького, к.п. н.,
доцент кафедри прикладної математики і
інформатики
e-mail: from4lu@gmail.com

О ПСИХОЛОГО-ПЕДАГОГИЧЕСКИХ АСПЕКТАХ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Аннотация. В статье рассматриваются психологические закономерности приема и обработки информации, функционирование психических приемов памяти и мышления у программиста, особенности развития личности студента в процессе обучения, существующие подходы к познавательной деятельности студентов.

Рассмотрены способы взаимодействия компонентов памяти, которые используются в работе программиста и к которым относят кратковременную память, долговременную память, рабочую память. Также описаны синтаксические и семантические аспекты, существующие в программировании.

В программировании важной составляющей является понимание программистом внутренней семантической структуры программы. В связи с этим автор рассматривает этот аспект процесса обучения студентов основам программирования. Также автором рассмотрены проблемы сложности содержания учебного материала и его объема.

Автор полагает, что акцентирование внимания на обозначенных аспектах в процессе обучения студентов основам программирования способствуют развитию и формированию профессиональных качеств будущих программистов.

Ключевые слова: учебно-познавательная деятельность студента, развитие личности студента, основы программирования, программист, психология программирования, компоненты памяти, семантические и синтаксические знания, содержание учебного материала.

Введение

Психологи различают два основных вида деятельности, связанные с познавательными процессами человека – *учение* и *обучение*. Согласно [1], "учение - целенаправленный процесс усвоения учащимися знаний, овладение умениями и навыками", "обучение – целенаправленный процесс передачи и усвоения знаний, умений, навыков и способов познавательной деятельности человека".

Польский ученый Ч. Куписевич считает, что обучение детерминировано целью, содержанием и действиями, с помощью которых субъект учится, приобретает

определенные знания, умения и навыки. Обучение разворачивается в результате собственной активности субъекта и носит, главным образом, процессуальный характер. Оно может протекать в различных формах, базируется на познании (прямом или косвенном) и индивидуальном опыте, вызывает изменения в поведении личности [2].

Различают две основные направленности процесса обучения: обучение понятиям и формированию умений (навыков) [3]. При обучении понятиям основное внимание уделяется представлению и разъяснению нового концептуального и фактического учебного материала: определению понятий, описания их свойств и отношений, обобщению и развернутому описанию процессов и т.п. При формировании умений главное внимание уделяется созданию проблемных ситуаций, которые повторяются и постепенно усложняются и требуют от студента выполнения определенных действий.

С другой стороны, Л. Фридман [4] выделяет следующие этапы учебной деятельности:

1) начально-мотивационный, который содержит следующие основные учебные действия:

- а) осознание учащимся основной учебно-проблемной ситуации, которая вводит его в предмет будущей работы по изучению учебного материала;
- б) формулировка основного учебного задания;
- в) самоконтроль и самооценка возможностей предстоящей деятельности по изучению учебного материала;

2) операционально-познавательный, на котором учащийся изучает учебный материал, овладевает знаниями, умениями и навыками в процессе выполнения основного учебно-познавательного задания;

3) контрольно-оценивающий, на котором учащийся обобщает изученный учебный материал, проводит корректировку ошибочных действий, анализирует работу, которая выполнена, оценивает свою деятельность в целом.

Основная часть

Условием успешного обучения в вузе есть интерес студентов к учебному предмету, ходу обучения и его результату. Этот интерес связан со многими факторами: содержанием учебного предмета, уровнем его сложности; организацией процесса обучения; системой контроля и оценки знаний; личностными качествами самого преподавателя; системой ценностей студента, его ближайшего окружения; взаимоотношениями в студенческом коллективе. Поэтому одной из важных задач педагогики высшей школы является активизация учебно-познавательной деятельности студентов ([5], [6] и др.). При этом целенаправленная деятельность преподавателя должна быть направлена на разработку и использование такого содержания, форм, методов, приемов и средств обучения, которые способствуют повышению познавательного интереса, активности, творческой самостоятельности студентов в усвоении знаний, формировании навыков и умений, применении их на практике [7]. В курсе по основам программирования творческий подход студентов к учебной деятельности формируется путем привлечения их к решению различных, постоянно усложняющихся, задач на лабораторных занятиях, во время самостоятельной работы, при подготовке рефератов и докладов, при выполнении курсовых работ, разработке коллективных проектов.

Точкой отчета начала повышенного интереса психологов к программированию является выход первого издания книги Вейнберга "The Psychology of Computer Programming" более 30 лет назад. Предметом исследования психологии программирования является деятельность программиста, психологические закономерности приема и обработки информации, функционирование психических

приемов памяти и мышления у программиста, его работоспособности, вопросы профессионального отбора.

В исследованиях J.G. Greeno [8], которые основаны на информационном подходе к психологии обучения, запоминания и решения задач, предложена схема взаимодействия компонентов памяти, которые используются в работе программиста и к которым относят кратковременную память, долговременную память, рабочую память. Согласно этому исследованию, постановка задачи (проблемы), которая должна быть запрограммирована, поступает в кратковременную память, в которой заранее анализируются сведения, поступившие предложения. В долговременной памяти находятся знания синтаксиса языка, методов построения алгоритмов, описания структур данных и т.п. В рабочей памяти сведения из кратковременной и долговременной памяти объединяются в новые структуры. В процессе решения задачи, т.е. генерации программы, новые сведения, хранящиеся в краткосрочной и долгосрочной памяти и понятия, относящиеся к проблеме, объединяются в рабочей памяти, а результат используется для генерации решения или, в случае обучения, запоминаются в долговременной памяти для дальнейшего использования. Как отмечено в [9]: "Хорошие программисты всегда немножко ленивы. Они сидят и ждут озарения, вместо того, чтобы воплощать в жизнь идею, что первой пришла им в голову". Это происходит благодаря тому, что опытный программист сохраняет в памяти сложные многоуровневые знания о понятиях и методах программирования и в процессе осмысления задачи определяет, какие из этих знаний следует применить. Часть этих знаний называют семантическими знаниями [10]. Эти знания относятся к общим для программирования понятиям и не связанным с конкретным языком программирования. К семантическим знаниям относят определение понятий нижнего уровня, например, как хранятся данные в памяти компьютера, как работает оператор ветвления, как происходит вызов подпрограммы, какие бывают типы данных. К ним относятся определения понятий среднего уровня, например, обмен значений двух переменных, алгоритм поиска большего из двух значений, а также более сложные алгоритмы, такие как алгоритмы сортировки, бинарный поиск, обход бинарного дерева поиска т.п. Эти знания хранятся в виде общих содержательных фреймов, в значительной степени не зависящих от знаний синтаксиса конкретных языков программирования или средств программирования, таких как пакеты подпрограмм или среды программирования.

В долговременной памяти сохраняются и синтаксические знания. Эти знания более конкретные и подробные, а потому легче забываются по сравнению с семантическими знаниями. К синтаксическим знаниям относятся такие элементы как синтаксис оператора присваивания, ветвления, повторения; зарезервированные слова; имена библиотечных функций и стандартных модулей.

Программистам, психологам и педагогам известно, что обычно трудно изучить первый язык программирования, а второй подобный изучить относительно легко потому, что легче выучить новое синтаксическое представление, если уже известна соответствующая семантическая структура.

Б. Шнейдерман указывает на различия между семантическими и синтаксическими знаниями в долговременной памяти программиста. Семантические знания содержатся в процессе, который является интеллектуальной потребностью осмысленного обучения, включая решение задач и изучения учебной литературы (учебников, справочников), что помогает закрепить новые понятия совместно с имеющимися семантическими знаниями. Синтаксические знания носят инструктивный характер, заучиваются механически и не очень пересекаются с уже имеющимися системами семантических знаний. Получению новых синтаксических знаний могут препятствовать полученные

ранее синтаксические знания, поскольку происходит добавление новых сведений, а не их интеграция. Исследования психологов подтверждают, что при изучении новых языков программирования программисты очень легко понимают семантику основных конструкций языка, но часто ошибаются при кодировании. Наш опыт работы показывает, что, например, студенты, уже имеющие опыт написания программ на языке Pascal, при программировании на языке C не ставят точку с запятой перед служебным словом "else" в операторе ветвления и не обращают внимание на различия идентификаторов, в которые входят одинаковые символы (буквы алфавита) в одной и той же последовательности, но в разных регистрах (строчные и прописные).

Исследователи предполагают, что процесс написания программы решения поставленной задачи состоит из двух фаз. На первой фазе сведения об условиях задачи поступают в систему мышления программиста и через кратковременную память попадают в рабочую память, где они анализируются. Аналогично из долговременной памяти программиста пересылаются обобщенные сведения (семантические и синтаксические) для дальнейшего анализа задачи. На второй фазе, при построении общего плана решения задачи, происходит ее пошаговое уточнение. Сначала решение задачи представляется в обобщенных терминах. Затем это представление развивается в более подробные планы до генерации конкретных кодов, отражающих самые мелкие детали.

Одной из характеристик профессиональных качеств программиста является понимание программ. Понимание программ необходимо при отладке и при модификации программ, а также при обучении. Б. Шнейдерман в своих исследованиях предполагает, что для понимания программы программист строит многоуровневую внутреннюю семантическую структуру, используя свои знания синтаксиса языка программирования. На самом высоком уровне программист должен понимать, какая задача решается с помощью программы. На более низких семантических уровнях программист может понимать знакомые последовательности операторов. Возможна ситуация, когда программист может понять детали нижнего уровня, не понимая общей задачи. "Главное заключается в том, что программисты производят внутреннюю семантическую структуру для представления синтаксиса программы, но не запоминают и не понимают программу как последовательность строк" [10]. Процесс запоминания, с помощью которого программист превращает программу во внутреннюю семантическую структуру, имеет "кусковой" характер [11]. Вместо того, чтобы разбирать программу символ за символом, программист распознает назначения групп операторов, затем собирает эти фрагменты вместе для формирования более крупных фрагментов, пока не будет понятна вся программа. Потому что внутренняя семантическая структура программы разрабатывается самым программистом, эти знания не быстро забываются и доступны для различных целей: использование других структур данных для решения задачи, написание программы решения задачи на другом языке программирования, разъяснения назначения программы кому-либо.

В процессе обучения будущих программистов также существуют синтаксические и семантические аспекты. Так, в курсах обучения второго или третьего (не первого) языка программирования одной парадигмы основное внимание уделяется синтаксическим аспектам семантических понятий. Курсы по разработке и анализу алгоритмов сосредоточены на семантических знаниях.

Согласно [10], семантические и синтаксические знания разделены. Многоуровневые семантические знания, полученные в основном благодаря целенаправленному обучению, находят свое воплощение в многоуровневом методе разработки внутренней семантики для некоторой конкретной задачи. Относительные синтаксические знания, приобретенные в основном благодаря механическому

заучиванию, полностью ограничены языком программирования. Семантические знания необходимы для анализа задачи, в то время как синтаксические знания используются на этапах кодирования и исполнения.

Многие ученые (Ю.И. Машбиц, Н.Ф. Талызина и др.) подчеркивают недопустимость возведения учебных целей только к усвоению предметных знаний и умений. На их взгляд в систему целей необходимо включать умение учиться. Мы полностью поддерживаем эту точку зрения и в этом контексте считаем, что в процессе обучения программированию надо определить систему задач, с помощью которой можно будет сформировать у студентов знания, умения и навыки, необходимые для их дальнейшего обучения и в следующей профессиональной деятельности. Система задач по программированию как средство обучения является одним из важнейших компонентов методического обеспечения в обучении основам программирования [12].

Заключение

Для обеспечения успешного обучения программированию необходимо решить вопрос о сложности содержания учебного материала и его объеме. Содержание учебного материала, с одной стороны, должен быть максимально приближенным к современному уровню развития компьютерных наук, с другой стороны, должно быть доступным студентам. Также надо систематизировать и определить последовательность его преподавания. Важно определить степень сложности учебного материала, так как слишком упрощенное или слишком сложное содержание ведут к снижению интереса и познавательной активности студентов в процессе обучения. Объем учебного материала не должен быть чрезмерным, чтобы не перегружать студентов в процессе обучения.

Так же важным является определение необходимого количества заданий для овладения учебным материалом и формирование практических умений и навыков. Ю.К. Бабанский отмечает, что "специальные исследования показали, что если придерживаться принципов систематичности и последовательности, то для формирования определенных умений и навыков достаточно выполнить пять упражнений при среднем уровне сложности учебного материала. Для определения количества упражнений в учебно-методических комплексах необходимо построение этих упражнений таким образом, чтобы на минимальном числе упражнений ученики овладевали бы общим принципом выполнения подобного рода действий" [13]. Возведение числа задач к минимуму приводит к тому, что их оказывается недостаточно для обеспечения целей обучения. С другой стороны, слишком большое количество задач ставит перед студентом непреодолимый барьер.

Отметим, что для обеспечения качественного усвоения студентами знаний по программированию следует обеспечивать не только наращивание числа исследуемых понятий, но и изучение понятий на разных уровнях детализации, научной строгости. Мы считаем, что дидактические принципы систематичности, последовательности обучения при построении курса по программированию целесообразно реализовывать в виде дидактической спирали - многократное возвращение на новом уровне [14]. Имеется в виду овладение студентами знаниями и умениями с помощью обогащения, развития и обобщения исследуемых понятий.

Список использованной литературы:

1. Український педагогічний словник / [авт.-уклад. Гончаренко С.]. – Київ : Либідь, 1997. – 376 с.
2. Куписевич Ч. Основы общей дидактики: Пер. с польск. О. В. Долженко. – М.: Высш. шк., 1986. – 368 с.
3. Экспертно-обучающие системы / Петрушин В.А.; Отв. ред. А.М. Довгяло; АН УССР, Ин-т кибернетики. – Киев: Наук. думка, 1992. – С. 196.
4. Фридман Л.М. Педагогический опыт глазами психолога: Книга для учителя. – М.: Просвещение, 1987. – 224 с.

5. Аврамчук Л.А. Формування активної пізнавальної діяльності студентів / Аврамчук Л.А. // Педагогіка і психологія. – 1997. – № 3. – С. 122–125.
6. Іваськів І.С. Активізація навчально-пізнавальної діяльності учнів на основі систем штучного інтелекту при навчанні інформатики в старшій школі: Дис. канд. пед. наук: 13.00.02 / НПУ ім. М.П. Драгоманова. – К., 2000. – 250 с.
7. Слєпкань З.І. Наукові засади педагогічного процесу у вищій школі. – К.: НТУ, 2000. – 210 с.
8. Greeno, J.G., The structure of memory and the process of problem solving, University of Michigan: Human Performance Center Technical Report 37, 1972. – p. 128.
9. Бенгли Дж. Жемчужины программирования / Бенгли Дж. ; [пер. с англ.]. – СПб. : Питер, 2002. – 272 с.
10. Шнейдерман Б. Психология программирования: Человеческие факторы в вычислительных и информационных системах. Пер. с англ.– М.: Радио и связь, 1984. – 304 с.
11. Miller, G.A., Needed: A better theory of cognitive organization, IEEE Transactions on Systems, Man, and Cybernetics, SMC-4, 1, (January 1974), p. 95-97.
12. Гришко Л.В. Система завдань з основ програмування як засіб підвищення якості навчання майбутніх програмістів // Інформаційні технології в освіті, науці і техніці. IV Всеукраїнська конференція молодих науковців ІТОНТ-2004. Черкаси, 2004. – С. 163-164.
13. Бабанский Ю.К. Дидактические проблемы совершенствования учебных комплексов //– М.: Просвещение, 1980. – С. 17–33.
14. Computing Curricula 2001. – [Электронный ресурс]: [Веб-сайт]. – Электронный ресурс. – Режим доступа: <http://www.computer.org/education/cc2009>– Название с экрана.

References:

1. Ukrainian pedagogical dictionary / [aut.-layout. Goncharenko C.]. - Kyiv: Lybid, 1997. - 376 p. (in Ukr)
2. Kupitsevich Ch. Fundamentals of general didactics: Per. from polsk O. Dolzhenko. - M. : Vyssh. Shk., 1986. - 368 p. (in Rus)
3. Experimental-teaching systems / Petrushin V.A.; Otv Ed. A.M. Dovgalyo; Academy of Sciences of Ukraine, Institute of Cybernetics. - Kiev: Science. opinion, 1992. - P. 196. (in Rus)
4. Friedman L.M. Pedagogical experience with the eyes of a psychologist: A book for a teacher. - Moscow: Enlightenment, 1987. - 224 p. (in Rus)
5. Avramchuk L.A. Formation of active cognitive activity of students / Avramchuk L.A. // Pedagogics and psychology. - 1997. - No. 3. - P. 122-125. (in Ukr)
6. Ivasky I.S. Activation of educational and cognitive activity of students on the basis of systems of artificial intelligence in computer science education in high school: Dis. Cand. ped Sciences: 13.00.02 / National Academy of Sciences of Ukraine M.P. Drahomanov - K., 2000. - 250 p. (in Ukr)
7. Slepkan Z.I. Scientific principles of pedagogical process in high school. - K. : NTU, 2000. – 210 p. (in Ukr)
8. Greeno, J.G., The structure of memory and the process of problem solving, University of Michigan: Human Performance Center Technical Report 37, 1972. – p. 128.
9. Bentley J. Pearls of Programming / Bentley J.; [per. with English.] - St. Petersburg : Peter, 2002. - 272 p. (in Rus)
10. Schneiderman B. Psychology of programming: Human factors in computing and information systems. Per. from English - M. : Radio and communication, 1984. - 304 p. (in Rus)
11. Miller, G.A., Needed: A better theory of cognitive organization, IEEE Transactions on Systems, Man, and Cybernetics, SMC-4, 1, (January 1974), p. 95-97.
12. Grishko L.V. System of tasks on the basis of programming as a means to improve the quality of training future programmers / Information technology in education, science and technology. IV All-Ukrainian Conference of Young Scientists ITONT-2004. Cherkasy, 2004. - P. 163-164. (in Ukr)
13. Babansky Yu.K. Didactic Problems of Improving Educational Complexes // - Moscow: Enlightenment, 1980. - pp. 17-33. (in Rus)
14. Computing Curricula 2001. – [Electronic resource]: [Website]. - Electronic data. - Access mode: <http://www.computer.org/education/cc2009>– Name from the screen.

GRYSCHKO Liudmyla,

The Bohdan Khmelnytsky National University of Cherkasy, PhD, Senior Lecturer

PSYCHOLOGICAL-PEDAGOGICAL ASPECTS OF PROGRAMMING TEACHING

Abstract. Introduction. The article discusses the psychological patterns of receiving and processing information, the functioning of the mental techniques of memory and thinking by a programmer, the characteristics of the development of the student's personality in the learning process, and the existing approaches to the cognitive activity of students.

Purpose. *The ways of interaction of memory components that are used in the programmer's work and which include short-term memory, long-term memory, working memory are considered. Syntactic and semantic aspects that exist in programming are also described.*

In programming, an important component is the understanding by the programmer of the internal semantic structure of the program. In this regard, the author considers this aspect of the process of teaching students the basics of programming. The author also examined the problem of complexity of the content of educational material and its volume.

Results. *The author believes that focusing on these aspects in the process of teaching students the basics of programming contribute to the development and development of professional skills of future programmers.*

Key words: *educational and cognitive activity of a student, development of a student's personality, basic programming, programmer, programming psychology, memory components, semantic and syntactic knowledge, content of educational material.*

Стаття надійшла 05.09.2017

Прийнято до друку 30.10.2017