

Газейкина Анна Ивановна,

кандидат педагогических наук, доцент, Институт математики, информатики и информационных технологий, Уральский государственный педагогический университет; 620075, г. Екатеринбург, ул. Карла Либкнехта, 9; e-mail: gazeykina@uspu.ru.

**ФОРМИРОВАНИЕ НАУЧНОГО МИРОВОЗЗРЕНИЯ БУДУЩЕГО ИТ-СПЕЦИАЛИСТА
В ПРОЦЕССЕ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ**

КЛЮЧЕВЫЕ СЛОВА: научное мировоззрение; теория и методика обучения программированию; стиль мышления; объектно ориентированное программирование; объектная декомпозиция.

АННОТАЦИЯ. Изучение программирования занимает важное место в предметной подготовке будущего ИТ-специалиста. В процессе освоения содержания этой дисциплины у студента формируется представление о технологии создания программных средств. Однако освоение программирования имеет и ярко выраженное развивающее и мировоззренческое значение. Изучение современных языков и технологий программирования способствует формированию специфических стилей мышления, а именно операционального, алгоритмического, объектного. Обучение программированию влияет на формирование научного мировоззрения будущего ИТ-специалиста. У студентов должно сформироваться представление о возможности двойственного взгляда на окружающую действительность – с точки зрения процессов (структурное программирование) и с точки зрения объектов (объектно ориентированное программирование). В статье предложены методические рекомендации по обучению студентов для наиболее полной реализации мировоззренческой функции курса программирования. Рекомендуется начинать обучение с объектно ориентированных языков программирования, начинать решение любой учебной задачи с построения объектной модели ее предметной области, анализировать историю развития языков программирования, рассматривать методологию объектно ориентированного программирования (основные идеи и принципы), применять в процессе обучения как визуальные, так и не визуальные среды программирования. Таким образом, структурное и объектно ориентированное программирование являются двумя инструментами моделирования окружающего мира вообще и предметной области решаемой задачи в частности.

Gazeykina Anna Ivanovna,

Candidate of Pedagogy, Associate Professor of Department of Informatics, Computer Technology and Methods of Teaching Informatics, Institute of Informatics and Information Technologies, Ural State Pedagogical University, Ekaterinburg, Russia.

**FORMATION OF SCIENTIFIC OUTLOOK OF THE FUTURE IT-SPECIALIST
IN TEACHING PROGRAMMING**

KEY WORDS: scientific outlook; theory and methods of teaching computer science; style of thinking; object-oriented programming; object-oriented decomposition.

ABSTRACT. Studying programming takes an important place in subject training of the future IT-specialist. In the course of study of programming, the students get the idea about the technology of software creation. However, studying programming has a strongly pronounced developing and world outlook value. Learning modern languages and programming technologies contributes to the formation of a particular style of thinking, namely the operational, algorithmic and object-oriented one. Training in programming affects the formation of the scientific world outlook of the future IT-specialist. Students should form a conception about the possibility of a dual view of the surrounding reality – in terms of processes (structured programming) and in terms of objects (object-oriented programming). The article suggests methodological recommendations for training students for the best realization of the worldview function of the course of programming. It is recommended to start training with object-oriented programming languages, to begin solving any teaching task with construction of an object model of its subject area, to analyze the history of programming languages, to consider the methodology of object-oriented programming (basic ideas and principles), and to use both visual and non-visual SDKs. So, structural programming and object-oriented programming are the two tools of modeling the world in general and the subject area of the problem, which is being solved, in particular.

Изучение программирования занимает важное место в предметной подготовке будущего ИТ-специалиста. В процессе освоения содержания этой дисциплины у студента формируется представление о технологии создания программных средств, без которых невозможны современные информационно-коммуникационные технологии. В настоящее время все чаще высказываются мнения об очень частной, сугубо предметной роли программиро-

вания в профессиональной подготовке будущего специалиста. Действительно, существование разнообразных программных средств информационных технологий делает, на первый взгляд, ненужным изучение программирования. Важно научить студента пользоваться этими готовыми, уже разработанными технологиями, научить его выбирать средства, адекватные решаемой задаче, и находить с помощью этих средств искомое решение. Заметим, что и сделать

это зачастую проще, чем обучать студентов технологиям создания программных средств.

Отметим также, что и в содержании школьного курса информатики (за исключением разве что профильных курсов) удельный вес раздела «Основы алгоритмизации и программирования» также существенно уменьшился.

С одной стороны, это справедливо. Несколько десятилетий назад программирование являлось по сути единственной технологией, применение которой позволяло получить решение поставленной задачи с помощью компьютера. Однако бурное развитие программного обеспечения избавило от необходимости программировать самому. Программирование стало узкоспециальным видом профессиональной деятельности.

Однако освоение программирования помимо узкопредметного и прикладного значения имеет и ярко выраженное развивающее и мировоззренческое значение. О влиянии изучения алгоритмизации и программирования на развитие мышления говорили еще классики информатизации образования А. П. Ершов, Ю. А. Первин, Г. А. Звенигородский, С. Пейперт и др., утверждая, что при изучении программирования формируется операциональное, алгоритмическое, логическое, абстрактное, комбинаторное и другие виды мышления. Зарубежные и отечественные психологи (Ж. Пиаже, Я. А. Пономарев, Д. А. Поспелов и др.) подтверждали это в своих исследованиях. При освоении современных языков и технологий программирования у студентов происходит формирование и развитие некоторых специфических стилей мышления, а именно операционального, алгоритмического, объектного [15, 6, 8].

В этой статье рассмотрим более подробно влияние обучения программированию на формирование научного мировоззрения будущего IT-специалиста.

Словари и энциклопедии трактуют мировоззрение как систему взглядов на объективный мир и место человека в нем, на отношение человека к окружающей его действительности и самому себе, а также как обусловленную этими взглядами основную жизненную позицию человека, его убеждения, идеалы, принципы познания и деятельности, ценностные ориентации. Мировоззрение – это предельное обобщение взглядов и представлений человека об окружающем мире, общее понимание мира, человека, общества, определяющее социально-политическую, философскую, религиозную, нравственную, эстетическую, научно-теоретическую ориентацию человека, его идеалы, убеждения, принципы познания, ценностные ориентации [12].

Под научным мировоззрением понимают мировоззрение, ориентирующееся в своих построениях на конкретные науки как на одно из своих оснований, особенно на их содержание – как материал для обобщения и интерпретации в рамках философской онтологии (всеобщей теории бытия) [11]. При этом сама наука как таковая мировоззрением в строгом смысле этого слова не является, так как, во-первых, она изучает саму объективную действительность, а не отношение человека к ней (а именно эта проблема является главным вопросом любого мировоззрения), а во-вторых, любое мировоззрение является ценностным видом сознания, тогда как наука – реализацией когнитивной сферы сознания, целью которой является получение знания о свойствах и отношениях различных объектов самих по себе. По мнению философов, особенно большое значение для научного мировоззрения имеет его опора на знание, получаемое в исторических, социальных и поведенческих науках, так как именно в них аккумулируется знание о реальных формах и механизмах отношения человека к действительности во всех ее сферах [15]. Однако несомненен вклад в формирование научного мировоззрения и естественных, и точных наук, в том числе и информатики.

Влияние изучения информатики (всех ее составляющих) на формирование научного мировоззрения обучаемого объясняется в первую очередь тем, что, по мнению современных ученых, вещество (материя), энергия и информация – это три важнейшие сущности нашего мира: из вещества мир состоит, энергией движется, информацией управляется. Эти фундаментальные понятия лежат в основе современной научной картины мира как целостной системы представлений об общих свойствах и закономерностях действительности [15].

Программирование, являясь одним из разделов информатики, также реализует мировоззренческую функцию.

В истории развития языков и технологий программирования выделяют несколько этапов:

- машинно ориентированные языки низкого уровня;
- языки высокого уровня;
- языки структурного программирования;
- объектно ориентированные языки.

Появление объектно ориентированных языков является следствием эволюционного развития языков программирования и отражает не только общие тенденции в развитии информационных технологий, но и общие подходы к познанию окружающего мира. Возникновение и развитие объектно

ориентированного подхода к созданию и использованию средств информационных технологий объясняют следующими событиями, причем не только в сфере программирования и информационных технологий:

- прогресс в области развития архитектуры ЭВМ;
- развитие языков программирования;
- развитие методологии программирования, включая принципы модульности и скрытия данных;
- развитие теории баз данных;
- исследования в области искусственного интеллекта;
- достижения философии и теории познания.

При изучении программирования в вузе у студентов должно сформироваться представление о возможности двойственного взгляда на окружающую действительность с точки зрения процессов (структурное программирование) и с точки зрения объектов (объектно ориентированное программирование). Еще древние греки высказывали идею о том, что мир можно рассматривать в терминах как объектов, так и событий, выделив таким образом существование и алгоритмической, и объектной декомпозиции. В XVII веке Декарт отмечал, что люди обычно имеют объектно ориентированный взгляд на мир. В XX веке эту тему развивала Рэнд в своей философии объективистской эпистемологии [14]. Марвин Мински предложил модель человеческого мышления, в которой разум человека рассматривается как общность различно мыслящих агентов – объектов [13]. Он доказывает, что только совместное действие таких агентов приводит к осмысленному поведению человека.

Алгоритмическая декомпозиция понимается как разделение алгоритмов, причем каждый модуль выполняет один из этапов общего процесса (профессор Э. Дейкстра [7]). Сущность объектно ориентированной декомпозиции состоит в разделении системы на элементы (объекты), где критерием разделения является принадлежность элементов к различным абстракциям (типам) данной предметной области. В объектно ориентированной декомпозиции мир задачи представляется совокупностью автономных действующих лиц (объектов), которые взаимодействуют друг с другом, чтобы обеспечить поведение системы, соответствующее более высокому уровню. Каждый объект модели обладает собственным поведением и моделирует некоторый объект реального мира, т. е. является вполне осязаемой «вещью», которая демонстрирует вполне определенное поведение. В отличие от алгоритмической декомпозиции, в объектно ориентированной модели не присут-

ствуют независимые алгоритмы, все действия существуют только как операции над определенными объектами, точнее над их полями. В объектно ориентированном программировании функциональный поток заменяется передачей сообщений между объектами, которые вызывают изменения состояния. Таким образом, объектно ориентированное программирование – это крайне естественный подход, поскольку структура программ непосредственно отражает структуру задачи. Более того, в моделируемых задачах обычно понятно, что является объектами. В частности, это могут быть машины на улице, механизмы производственной линии, геометрические фигуры и т. д.

Мировоззренческое значение освоения объектно ориентированного программирования заключается в реализации следующей связи между понятиями: объект – источник информации – изучение объекта через его свойства – изменение свойств объекта через его методы – поведение объекта при взаимодействии с внешней средой [15].

Возникает вопрос: какая декомпозиция сложной системы рациональнее – по алгоритмам или по объектам? Анализ обозначенной проблемы дает возможность утверждать, что важны оба аспекта. Но систему невозможно сконструировать сразу двумя ортогональными способами: следует начинать с объектной декомпозиции, так как она имеет следующие преимущества перед алгоритмической:

- позволяет повторно использовать общие механизмы, что приводит к существенной экономии выразительных средств, а также уменьшает размер программы;
- объектно ориентированная система более гибка, проще изменяется и эволюционирует;
- объектная декомпозиция помогает разобраться в сложной системе, предлагая, как правило, более разумные варианты решения поставленной задачи.

После объектной декомпозиции, используя полученную структуру, следует рассмотреть проблему и с другой точки зрения – алгоритмической. Таким образом, объектно ориентированная декомпозиция не отрицает декомпозицию алгоритмическую, а включает ее в себя, подчиняя построенной объектной модели.

Для наиболее полной реализации мировоззренческой функции курса программирования в вузе можно сформулировать следующие рекомендации по построению содержания курса и применению методов обучения и методических приемов.

Начинать обучение программированию следует не со структурных, а с объектно ориентированных языков программирова-

ния (Object Pascal, Java, C++) [2, 4, 5]. На начальном этапе освоения объектно ориентированного программирования классы, которые разрабатывают студенты, могут (и должны) быть простейшими: несколько полей (элементарных, а не структурированных типов), конструктор и несколько методов, реализующих линейные алгоритмы обработки данных (полей).

На начальном этапе важно сформировать у обучаемых представление о классе как совокупности полей и методов, научить их объявлять класс как модель некоторой предметной области задачи и создавать объекты классов. При этом важны как синтаксическая, так и семантическая сторона используемых конструкций языка программирования. Студент должен не просто запомнить правила синтаксиса, но осознать их смысловую обусловленность [1, 4, 8].

На этом этапе можно придерживаться следующих методических рекомендаций:

- при решении любой учебной задачи перед реализацией класса на конкретном языке программирования сначала спроектировать его, а именно:

- выделить поля и определить их типы (у обучаемого должно сформироваться четкое представление о том, что поля – это данные об объекте, они являются независимыми (если некоторое данное зависит от другого, то это уже не поле, а метод – функция), и вспомогательные (локальные) переменные не являются полями);

- продумать формальные параметры для конструктора (или конструкторов) класса;

- на основе поставленной задачи выделить методы класса, обязательно указав, какие из них будут являться модификаторами (т. е. будут изменять значения полей), а какие – селекторами (т. е. будут сообщать о состоянии полей); особое внимание при этом необходимо уделить формированию представления о методе как подпрограмме, инкапсулированной в класс, что будет вступать в некоторое противоречие с усвоенными ранее (например, при изучении структурного программирования в школьном курсе информатики) представлениями, в частности, о формальных параметрах методов;

- проследивать жизненный цикл объекта от момента его создания с помощью конструктора до удаления из памяти с помощью деструктора (если он имеется);

- предлагать студентам обратные задачи: на основе представленного описания класса предложить возможные варианты предметной области, которая может быть представлена с помощью такого описания.

Следующим этапом обучения является освоение основных алгоритмических кон-

струкций и их реализации в конкретном языке программирования, а также системы его типов данных. По сути, при этом рассматривается структурное программирование, без него невозможно реализовать классы для решения более сложных задач. При этом рекомендуется:

- применять структурированные типы данных (в первую очередь, массивы) как для организации полей класса, так и для организации объектов класса;

- аналогичным образом использовать базовые алгоритмические конструкции как при описании методов, так и для манипулирования объектами класса;

- обсуждать со студентами возможность двойственного взгляда на окружающую действительность: с точки зрения процессов и с точки зрения взаимодействующих объектов; самое простое задание: представьте учебное занятие как процесс, т. е. алгоритм, имеющий начало, реализацию базовых алгоритмических конструкций (линейной, ветвления, цикла) и конец, и как взаимодействие объектов, принадлежащих различным классам: люди (студенты и преподаватель), компьютерная техника, средства обучения и т. п.

Затем целесообразно рассмотреть различные отношения между классами, в первую очередь – наследование, а затем и композицию классов. При этом решение любой учебной задачи необходимо начинать с построения прежде всего объектной модели ее предметной области, и только потом алгоритмической.

Представляется целесообразным анализировать на занятиях историю развития языков программирования, раскрывать студентам сущность понятия «парадигма программирования».

Автор рекомендует рассматривать технологию объектно ориентированного программирования на методологическом уровне (основные идеи и принципы), а не только на прикладном. Содержание понятий «объектно ориентированный метод» и «объектно ориентированный подход» существенно шире, чем просто «объектно ориентированное программирование». Они включают философию разработки систем программирования, извлечение знаний, анализ требований, моделирование предметной области, проектирование систем, проектирование баз данных и многие другие вопросы. Конечно, в начальном курсе программирования не найдет воплощения указанная философия и ее использование для решения проблем, возникающих при создании информационных систем. Однако рассмотрение этих вопросов и иллюстрация их конкретными примерами позволит в даль-

нейшем будущему ИТ-специалисту применить эти теоретические знания в профессиональной деятельности, быть в курсе актуального состояния языков и технологий программирования. Именно поэтому целесообразно анализировать различия в содержании понятий «объектно ориентированное» (object-oriented), «компонентно ориентированное» (component-oriented), «основанное на объектах» (object-based) и «основанное на классах» (class-based) программирование, а также в их разработке и анализе.

Рассматриваемые методологические основы объектно ориентированного подхода можно закрепить при выполнении учебных заданий [3, 9], подобных приведенным ниже.

Задание 1. Какая концепция искусственного интеллекта является наиболее близкой к понятию объекта?

- а) слот
- б) механизм вывода
- в) база знаний
- г) фрейм
- д) фацет
- е) правило

Задание 2. Чем различаются перечисленные ниже понятия (сформулируйте по одному предложению для каждого пункта)?

- а) экземпляр и класс
- б) тип данных и класс
- в) класс и роль
- г) тип объекта и тип сущности
- д) класс и компонент
- е) динамическое связывание и полиморфизм
- ж) обобщение и наследование
- з) наследование и классификация

Задание 3. Прокомментируйте следующее высказывание: «Объектная технология умерла. Ей на смену пришла компонентно ориентированная разработка.»

Выполнение подобных заданий будет способствовать формированию у студентов обобщенных взглядов и представлений об окружающем мире [3, 10], его общему пониманию, а значит, и формированию научного мировоззрения.

В качестве еще одной рекомендации сформулируем целесообразность применения в процессе обучения как визуальных, так и невизуальных сред программирования [2, 4, 5]. Это позволит сформировать у студентов представление о независимости результатов решения задачи от используе-

мого инструментария, умение использовать различные инструменты для решения сходных задач, а также умение осваивать новое программное обеспечение.

В рамках данной статьи представляет интерес анализ курсовых работ, выполненных студентами второго курса Института математики, информатики и информационных технологий Уральского государственного педагогического университета. Курсовые работы выполнялись по дисциплинам «Технологии программирования», «Языки и технологии программирования» (в зависимости от направления подготовки студентов). Около 30% студентов при выполнении курсовой работы самостоятельно освоили новый для них инструментарий: языки, среды программирования, библиотеки, интерфейсы и т.п. Подавляющее большинство студентов (более 80%) грамотно выполнили объектно ориентированный анализ предметной области и объектно ориентированное проектирование, представили методологически правильно разработанный и устойчиво функционирующий программный продукт. Конечно, эти результаты в первую очередь свидетельствуют о качестве предметной подготовки будущих ИТ-специалистов в области программирования. Но также они говорят о том, что студенты овладели некоторыми способами и инструментами моделирования окружающего мира, что оказало влияние и на их научное мировоззрение.

Современное миропонимание – важный компонент человеческой культуры. Очевидно, что каждый культурный человек должен представлять, как устроен мир, в котором он живет. Изучение объектно ориентированного программирования дает возможность рассмотреть окружающий мир с двух разных точек зрения: как совокупность объектов и как совокупность процессов. Структурное и объектно ориентированное программирование являются двумя инструментами моделирования окружающего мира вообще и предметной области решаемой задачи в частности. Следовательно, изучение программирования активно участвует в формировании мировоззрения человека – совокупности его обобщенных взглядов на мир в целом и на свое отношение к этому миру.

ЛИТЕРАТУРА

1. Алексеевский П. И. Применение средств управления версиями для коллективной работы студентов над проектом компьютерной игры. // Педагогическое образование в России. 2012. № 6. С. 51–54.
2. Алексеевский П. И., Лапенков М. В. Выбор программного обеспечения для проведения практических занятий по программированию на С и С++ // Информатика и образование. 2010. № 2. С. 117–119.
3. Газейкина А. И. Обучение будущего учителя информатики конструированию учебных заданий, направленных на формирование метапредметных результатов обучения // Педагогическое образование в России. 2014. № 8. С. 159–164.

4. Газейкина А. И. Обучение программированию будущего учителя информатики // Педагогическое образование в России. 2012. № 5. С. 45–48.
5. Газейкина А. И. Обучение программированию будущих ИТ-специалистов с применением дистанционных технологий // Подготовка молодежи к инновационной деятельности в процессе обучения физике, математике, информатике : материалы международной научно-практической конференции. Урал. гос. пед. ун-т. Екатеринбург, 2014. С. 33–37.
6. Газейкина А. И. Стили мышления и обучение программированию студентов педагогического вуза. URL: <http://ito.edu.ru/2006/Moscow/I/1/I-1-6371.html>.
7. Дейкстра Э. Дисциплина программирования. М. : Мир, 1978.
8. Рожина И. В. Обучение учащихся объектно-ориентированному программированию и технологии визуального проектирования в базовом курсе информатики : дис. ... канд. пед. наук. Екатеринбург, 2002.
9. Семенова И. Н., Кузьмина Т. А. Конвенционально-ролевая рефлексия как механизм проявления автотологичности методов обучения в процессе педагогического образования студентов // Педагогическое образование в России. 2012. № 2. С. 151–154.
10. Усольцев А. П., Курочкин А. И. Концепция развивающего обучения при построении системы задач как средство решения современных образовательных проблем // Педагогическое образование в России. 2013. № 6. С. 248–251.
11. Философия науки: словарь основных терминов. М. : Академический Проект. С.А. Лебедев. 2004.
12. Философия : энциклопедический словарь. М. : Гардарики. Под редакцией А.А. Ивина. 2004.
13. Minsky M. The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. – Simon & Schuster Paperbacks, 2007.
14. Rand A. Introducing Objectivism, in Peikoff, Leonard, ed. The Voice of Reason: Essays in Objectivist Thought. Meridian, New York, 1990.
15. Starichenko B. E. Conceptual basics of computer didactics : monograph. Yelm, WA, USA: Science Book Publishing House, 2013.

L I T E R A T U R E

1. Alekseevskiy P. I. Primenenie sredstv upravleniya versiyami dlya kollektivnoy raboty studentov nad proektom komp'yuternoy igry. // Pedagogicheskoe obrazovanie v Rossii. 2012. № 6. S. 51–54.
2. Alekseevskiy P. I., Lapenok M. V. Vybora programmnogo obespecheniya dlya provedeniya prakticheskikh zanyatiy po programmirovaniyu na C i C++ // Informatika i obrazovanie. 2010. № 2. S. 117–119.
3. Gazeykina A. I. Obuchenie budushchego uchitelya informatiki konstruirovaniyu uchebnykh zadaniy, napravlenykh na formirovaniye metapredmetnykh rezul'tatov obucheniya // Pedagogicheskoe obrazovanie v Rossii. 2014. № 8. S. 159–164.
4. Gazeykina A. I. Obuchenie programmirovaniyu budushchego uchitelya informatiki // Pedagogicheskoe obrazovanie v Rossii. 2012. № 5. S. 45–48.
5. Gazeykina A. I. Obuchenie programmirovaniyu budushchikh IT-spetsialistov s primeneniem distantsionnykh tekhnologiy // Podgotovka molodezhi k innovatsionnoy deyatel'nosti v protsesse obucheniya fizike, matematike, informatike : materialy mezhdunarodnoy nauchno-prakticheskoy konferentsii. Ural. gos. ped. un-t. Ekaterinburg, 2014. S. 33–37.
6. Gazeykina A. I. Stili myshleniya i obuchenie programmirovaniyu studentov pedagogicheskogo vuza. URL: <http://ito.edu.ru/2006/Moscow/I/1/I-1-6371.html>.
7. Deykstra E. Distsiplina programmirovaniya. M. : Mir, 1978.
8. Rozhina I. V. Obuchenie uchashchikhsya ob"ektно-orientirovannomu programmirovaniyu i tekhnologii vizual'nogo proektirovaniya v bazovom kurse informatiki : dis. ... kand. ped. nauk. Ekaterinburg, 2002.
9. Semenova I. N., Kuz'mina T. A. Konventsial'no-rolevaya refleksiya kak mekhanizm proyavleniya avtologichnosti metodov obucheniya v protsesse pedagogicheskogo obrazovaniya studentov // Pedagogicheskoe obrazovanie v Rossii. 2012. № 2. S. 151–154.
10. Usol'tsev A. P., Kurochkin A. I. Kontseptsiya razvivayushchego obucheniya pri postroenii sistemy zadach kak sredstvo resheniya sovremennykh obrazovatel'nykh problem // Pedagogicheskoe obrazovanie v Rossii. 2013. № 6. S. 248–251.
11. Filosofiya nauki: slovar' osnovnykh terminov. M. : Akademicheskii Proekt. S.A. Lebedev. 2004.
12. Filosofiya : entsiklopedicheskiy slovar'. M. : Gardariki. Pod redaktsiey A.A. Ivina. 2004.
13. Minsky M. The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. – Simon & Schuster Paperbacks, 2007.
14. Rand A. Introducing Objectivism, in Peikoff, Leonard, ed. The Voice of Reason: Essays in Objectivist Thought. Meridian, New York, 1990.
15. Starichenko B. E. Conceptual basics of computer didactics : monograph. Yelm, WA, USA: Science Book Publishing House, 2013.

Статью рекомендует д-р пед. наук, профессор Б. Е. Стариченко.