

НЕКОТОРЫЕ СОДЕРЖАТЕЛЬНЫЕ ПРОБЛЕМЫ ПРЕПОДАВАНИЯ В ВУЗАХ ТЕОРИИ ОТКРЫТЫХ СИСТЕМ

SOME SUBSTANTIVE PROBLEMS OF TEACHING OPEN SYSTEMS THEORY IN UNIVERSITIES

**R. Fokin
M. Abissova
E. Ershova**

Summary: The article substantiates the expediency of teaching the theory of open systems (TOS) in universities when teaching computer science and, in the future, also technical and pedagogical sciences. That TOS was developed by IBM in the 1960s. Since then, this theory has been the core of modern computer science. Open systems (OS) are understood by IBM as systems for which there are open (accessible to all) descriptions (specifications) of their internal and external interfaces. An interface refers to a certain communication module. Internal connections are the connections between the components of the system. External links of this system with others. Are discussed: the theoretical foundations of TOS and approaches to its teaching; features of programming in TOS and OS Linux; the history of the emergence and development of TOS; some limits of TOS application; examples of the spread of TOS to technical and pedagogical sciences.

Keywords: students, learning, induction, deduction, historicism, computer science, open systems, specifications, stratification, metamodeling, technology, pedagogy.

Фокин Роман Романович

Доктор педагогических наук, доцент, Федеральное Государственное Бюджетное Военное Образовательное Учреждение Высшего Образования «Военно-космическая академия имени А.Ф. Можайского», (г. Санкт-Петербург)
vka@mil.ru

Абиссова Марина Алексеевна

Кандидат педагогических наук, доцент, Федеральное Государственное Бюджетное Образовательное Учреждение Высшего Образования «Государственный университет морского и речного флота имени адмирала С.О. Макарова», (г. Санкт-Петербург)
tarabyss@yandex.ru

Ершова Елена Викторовна

Преподаватель, Федеральное Государственное Бюджетное Военное Образовательное Учреждение Высшего Образования «Военно-космическая академия имени А.Ф. Можайского», (г. Санкт-Петербург)
vka@mil.ru

Аннотация: Статья обосновывает целесообразность преподавания теории открытых систем (ТОС) в вузах при обучении информатике и в перспективе также техническим и педагогическим наукам. ТОС была разработана IBM в 1960 годы. С тех пор эта теория является ядром современной информатики. Открытые системы (ОС) понимаются IBM как системы, для которых существуют открытые (доступные всем) описания (спецификации) внутренних и внешних их интерфейсов. Под интерфейсом понимается некоторый модуль связи. Внутренние связи – это связи между компонентами системы. Внешние – связи данной системы с другими. Обсуждаются: теоретические основы ТОС и подходы к ее преподаванию; особенности программирования в ТОС и ОС Linux; история возникновения и развития ТОС; некоторые пределы применения ТОС; примеры распространения ТОС на технические и педагогические науки.

Ключевые слова: студенты, обучение, индукция, дедукция, историзм, информатика, открытые системы, спецификации, стратификация, метамоделирование, техника, педагогика.

Введение в проблематику исследования

Актуальность статьи объясняется тем, что ТОС как дисциплина или раздел в составе информатики студентам обычно не преподаются. О распространении ТОС на другие области знаний им также не рассказывается, хотя имеются соответствующие научные обобщения и опыт.

Цель исследования: Предложить некоторые аспекты содержания обучения ТОС. Важность системного применения этих аспектов на практике дает обоснование необходимости преподавания ТОС современным студентам при изучении ими информатики, а в перспективе – других областей знаний.

Задачи: рассмотреть следующие аспекты содержания обучения ТОС:

1. теоретические основы ТОС и подходы к ее преподаванию;
2. особенности программирования в ТОС и ОС Linux;
3. историю возникновения и развития ТОС;
4. некоторые пределы применения ТОС;
5. примеры распространения ТОС на технику и педагогику.

1. Теоретические основы ТОС и подходы к ее преподаванию

Рассмотрим ОС в смысле, введенном IBM. Пусть изна-

чально некоторый производитель выпускает такую систему, и пусть она имеет большой успех у потребителей. Пусть для некоторого ее компонента имеются открытые спецификации [1, 2], описывающие, что у него должно быть на входе и на выходе. Тогда другие производители смогут разрабатывать аналоги этого компонента с теми же входом и выходом, но физически эти аналоги могут быть реализованы иначе. Представители этих производителей могут организовывать рабочие группы по совершенствованию открытых спецификаций по мере развития технологий. Начальный производитель (за счет им же открытых для всех спецификаций) получает всемирную популярность у потребителя, стабильные сбыт продукции и доходы на протяжении десятилетий, идеологическое влияние и управление большим числом производителей.

Полностью открытая система – очень редкое явление, как и полностью закрытая. Обычно не все интерфейсы имеют открытые спецификации. Удачные открытые (даже частично) системы обычно живут очень долго, постоянно совершенствуясь. Например, редактор Microsoft Word имеет версии 1993, 1995, 2000, 2007, 2010, 2016, 2019, 2021. Умирают лишь неудачные решения. А пользователю проще перейти с Microsoft Word 2019 на его версию 2021, чем на совсем другой редактор. При этом минимизируются отрицательные последствия нововведений.

При разработке некоторой ОС необходимо разработать не просто изделие, а развивающуюся платформу, способную к перманентному совершенствованию. Всякое изделие производится на основе его моделей. Для ОС разрабатывается [1, 2] стратифицированная метамодель. Страта – это некоторый слой слоистых облаков, понятие пришло из метеорологии. Метамодель – это модель моделей, модель более высокого уровня абстракции, она обычно более устойчива к изменениям в условиях прогресса науки.

Стратифицированная метамодель – это совокупность связанных моделей, находящихся на различных уровнях абстракции. Модели уровня 0 непосредственно обеспечивают производство изделия. Модели уровня 1 – это модели более высокого уровня абстракции, чем модели уровня 0. Модели уровня 2 – это модели еще более высокого уровня абстракции и так далее. Идея стратифицирования формальной теории принадлежит Ноаму Хомски – величайшему математику нашего времени, специалисту в области теории формальных грамматик и языков. Совокупность открытых спецификаций тоже представляет собой стратифицированную по уровням абстракции систему.

Ниже будет показано, что при обучении ТОС необходимо значимо использовать индуктивные подходы (когда

из частных случаев выводится общее заключение), а не только дедуктивные (когда из общего суждения выводятся частные случаи). При этом частные случаи обычно представляют собой исторические факты, относящиеся к истории информатики и техники, тем самым в преподавании ТОС значимым является исторический подход. Даже среди студентов следующих направлений: информационные технологии (ИТ), техника, математика, физика, естественные науки доля правополушарных [1, 3] превышает сейчас 2/3 и растет, что делает актуальным применение методов индукции и историзма при обучении.

2. Особенности программирования в ТОС и ОС Linux

Пусть имеется готовая отлаженная процедура OldSub (OldInput, OldOutput), где OldInput и OldOutput – последовательности параметров, заполняемые данными при входе в процедуру и при выходе из нее соответственно, пусть эти входные и выходные параметры полностью описаны открытой спецификацией. Разбираться в смысле операторов (команд), которые непосредственно реализуют эту процедуру очень трудоемко, также они могут быть скрытыми для просмотра.

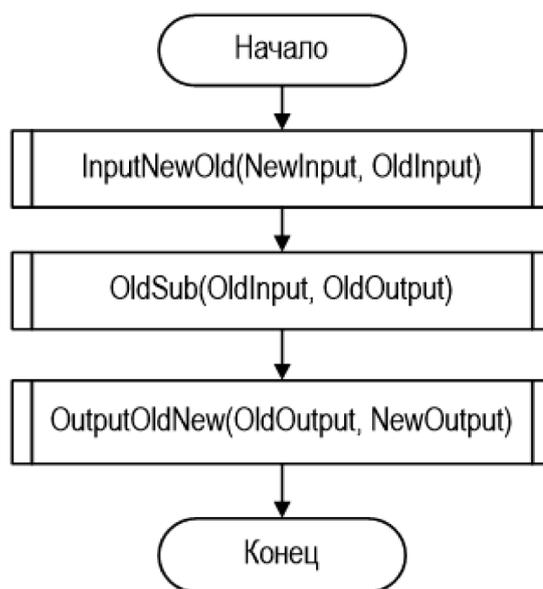


Рис. 1. Блок-схема NewSub (NewInput, NewOutput)

Нужно разработать немного иную процедуру NewSub (NewInput, NewOutput), где NewInput и NewOutput – последовательности параметров, заполняемые данными при входе в процедуру и при выходе из нее соответственно, эти входные и выходные параметры полностью описаны открытой спецификацией.

Рис. 1 показывает, как можно реализовать процедуру NewSub малыми трудозатратами на программирование. Достаточно разработать несложные процедуры InputNewOld и OutputOldNew. Первая по NewInput (входным

параметрам NewSub) формирует параметры OldInput для последующей подстановки в OldSub. Вторая по OldOutput (выходным параметрам OldSub) формирует выходные параметры NewInput.

Указанный выше прием для NewSub дает менее эффективный код (в смысле и времени исполнения, и размера результирующего файла), чем можно было бы получить, программируя «с чистого листа» или разбираясь в деталях алгоритма OldSub. Если такие модификации производить многократно, то программный код становится существенно менее эффективным. Пользователи замечают, что при исполнении идентичных команд старый Microsoft Word 1993 «летал», а новый Microsoft Word 2021 «ползает». Приходится применять все более мощные процессоры.

Фирма Apple только последние лет десять частично использует технологии ОС, что существенно удешевило ее продукцию и подняло ее конкурентоспособность. Несмотря на это, по качеству программного обеспечения (ПО) Apple до сих пор превосходит большинство других разработчиков. В результате компьютеры, планшеты, смартфоны Apple используют значительно менее мощные аппаратные средства (АС) – процессоры в основном, которые меньше нагреваются, более надежны, и при этом их пользователи не жалуются на «тормоза».

Классической полностью открытой системой [1] является семейство операционных систем Linux. Линус Торвалдс (родился в 1969 в Хельсинки) самостоятельно разработал новую операционную систему Linux и в 1991 выложил в открытый для всех файл-обменник в интернете не только ее исполнимые программные файлы, но и их исходные тексты. Исходные тексты – это тексты на языках программирования. В настоящее время из-за объектно-ориентированного программирования (и, как следствие, визуально и событийно-ориентированного) чаще говорят об исходных проектах. Не все файлы исходного проекта имеют текстовые форматы.

Он предложил всем желающим бесплатно пользоваться Linux, сообщать о замеченных ошибках в ее работе, вносить любые изменения в исходные тексты для ее совершенствования, разрабатывать ПО под Linux. Л. Торвалдс относит себя к сообществу, выступающему за свободное распространение информации. Этим сообществом были разработаны и юридически закреплены правила для пользователей и разработчиков Linux, запрещающие коммерциализацию последней и ограничивающие коммерциализацию ПО, разработанного под нее.

В настоящее время развивается несколько направлений Linux – Arch, Fedora, Debian, Ubuntu, Red Hat Enterprise, CentOS и многие другие. В отличие от не полностью открытых Windows и ПО под нее, Linux и ПО под него (из-

за полностью открытых исходных проектов) позволяют убедиться (приложив немалые усилия) в отсутствии внутри вредоносных закладок.

3. История возникновения и развития ТОС

В 1930 годы Л. фон Бергаланфи предложил общую теорию систем – философскую, по сути, дисциплину, где ОС были определены как системы, которые постоянно обмениваются веществом и энергией с внешней средой. Такое понимание ОС не обсуждается в данной статье. Как указано в аннотации, обсуждаются ОС в смысле, введенном IBM.

В середине 1960 годов IBM [4] предложила систему 360 – линейку универсальных по назначению компьютеров открытой архитектуры [4, 5] различной мощности и стоимости. Архитектура компьютера – это битовые структуры типов данных (обрабатываемых им) и команд (исполняемых им). Система 360 – это первая в истории ОС в смысле IBM.

Разработав в начале 1970 годов серию ЕС ЭВМ ряд 1, СССР скопировал лишь то, что мог – открытую архитектуру. Элементная база (ЭБ) открытой не была. ЭБ серии ЕС ЭВМ была отечественной, по ЭБ СССР отставал от США на 5-7 лет. Затем появились аналоги более совершенной системы 370 – ЕС рядов 2, 3 и 4. Ввиду идентичности архитектуры имела место полная программная совместимость IBM system 360/370 и ЕС ЭВМ рядов 1-4. Путем файлового копирования СССР получил для ЕС ЭВМ большое количество ПО – системного, инструментального, прикладного. В СССР и других странах социализма за 15-20 лет было выпущено около 15-20 тысяч ЕС ЭВМ. Все наши крупные вузы и предприятия тех лет обычно имели свои ЕС ЭВМ.

Вслед за IBM по пути ОС пошли другие производители компьютеров (например, DEC и HP), а в СССР появились мини-ЭВМ серии СМ и микро-ЭВМ серии ДВК. Затем ОС Intel и IBM породили у нас персональные компьютеры (ПК) серий ЕС и Искра. В 1993 году в России появилось научное издательство «Открытые системы», его журналы описывают приложения ОС к информатике, чаще – к ПО, реже – к АС.

В настоящее время часто высказывается мнение, что решение правительства СССР о копировании открытых архитектур компьютеров IBM, DEC, HP было ошибочным. Якобы альтернативное решение об отказе от использования зарубежных открытых архитектур и развитии отечественных направлений ЭВМ привело бы в итоге к лучшему для СССР результатам. При этом отечественную ЭВМ БЭСМ-6 считают наиболее перспективной для такого развития. Гостивший в Ленинграде в конце 1970 годов Эдсгер Дейкстра якобы говорил, что факт производства

в СССР компьютеров IBM означает крупнейшую победу США в холодной войне. Говорят, что эта мысль взята из его речи в здании Академии наук на Васильевском острове.

Авторы настоящей статьи считают, что указанное выше решение правительства СССР было верным. Открытая архитектура компьютеров IBM system 360/370 (они производились с 1964 и с 1970 годов соответственно) содержит решения фундаментального характера, использующиеся почти во всех современных компьютерах – читайте аннотацию, а также раздел 1 настоящей статьи. Например, поле целого числа могло иметь длину 1 байт (8 битов), 2, 4 или 8 байтов, поле действительного числа с плавающей точкой – 4 или 8 байтов. Использовались 32-битовые адреса байтов памяти как беззнаковые целые, 2-байтовые поля адресовались беззнаковыми четными числами, 4-байтовые поля – беззнаковыми целыми, кратными 4, 8-байтовые поля – беззнаковыми целыми, кратными 8, имеется страничная организация памяти. Имелись сотни 1-адресных и 2-адресных команд для работы с действительными и целыми числами различной длины. Таким образом, компьютеры IBM system 360/370 относились к типу больших универсальных компьютеров (но не супер-компьютеров), способных эффективно решать расчетные задачи (научно-технические с плавающей точкой, финансовые – целочисленная арифметика с 8-байтовыми полями обеспечивала расчет триллионов долларов с точностью до цента), а также – логические (обеспечивались развитой целочисленной арифметикой). Это были компьютеры 3 поколения (элементная база – большие интегральные схемы, БИС). В СССР БИС освоили к началу 1970 годов, а в 1980 годы – сверх-БИС (СБИС) для ЭВМ 4 поколения.

БЭСМ-6 относилась ко 2 поколению (ЭБ – транзисторы) и к типу супер-компьютеров. Она была очень дорогой, производилась с 1968 по 1987 год. Всего было произведено лишь 355 экземпляров. Начало производства ЕС ЭВМ не означало конца производства БЭСМ-6. Таким образом, в 1987 году в СССР одновременно с БЭСМ-6 (2 поколения) производились уже компьютеры 4 поколения, включая ПК! Супер-компьютеры ориентированы лишь на расчеты с плавающей точкой, они не универсальны. Супер-компьютеры никогда ни в какой стране ОС не являлись, поэтому развитие технологий ОС в СССР на базе БЭСМ-6 было бы невозможным. БЭСМ-6 имела всего 52 команды, причем только 1-адресные, именно за счет этого она имела высокое быстродействие. Процессор БЭСМ-6 предвосхитил современную идею RISC-процессоров (имеющих простую архитектуру и высокое быстродействие за счет этого), которые также используются на современных супер-компьютерах. БЭСМ-6 обрабатывала и адресовала лишь 48-битные поля данных, адреса были 15-битные, хотя присутствовала страничная

организация памяти. Такая специфическая архитектура с 15-битными адресами, с 48-битными полями данных исключительно, с 48-битным форматом действительного числа с плавающей точкой умерла в 1987 году.

Некоторым авторам настоящей статьи [6] приходилось заниматься низкоуровневым (машинно-ориентированным) программированием в кодах процессора и на ассемблере. Для этого необходимо знать архитектуру конкретного компьютера. Те, с кем им приходилось говорить, кто утверждал ошибочность обсуждаемого здесь решения правительства СССР и отстаивал развитие в СССР технологий ОС на базе БЭСМ-6, занимались лишь высокоуровневым (машинно-независимым) программированием, соответственно, они и не представляли себе в должной степени сущности архитектуры компьютера.

Э. Дейкстра известен работами по теоретическому программированию и теории алгоритмов, архитектурой компьютера он, вероятно, никогда не занимался. Мысль (якобы принадлежавшая ему), приведенная выше в данном разделе, авторами статьи в виде прямой речи не была найдена ни в обычной литературе, ни в интернете. Возможно, она им никогда и не высказывалась, а была ему безосновательно приписана.

4. Некоторые пределы применения ТОС

Речь идет о сложных системах. Макро и микро-методы – это 2-уровневая модель методов работы с ними. ТОС может предложить для методов и большее число уровней. Но для применения ТОС важны пределы масштабирования.

С 1980 годов до настоящего времени [4] ЭБ компьютеров – это СБИС, СБИС – это электрическая (электронная) схема, состоящая из компонентов, описанных в учебниках физики. Но в СБИС их миллиарды. Человек может СБИС проектировать и реализовывать, используя специальные инструменты, ПО в частности. Но ход физических явлений не будет неизменным при масштабировании в сторону многократного уменьшения размеров компонентов схемы и увеличения их количества. Квантовая, фотонная, органическая виды электроники пока не реализованы материально. К 2000 году тактовая частота процессоров достигла 2 ГГц и с тех пор почти не растет.

В рамках «Японского проекта» [7] планировался в 1990 годы переход на компьютеры 5 поколения. Для них задачи искусственного интеллекта (ИИ) должны были стать базовыми (вместо вычислительных), язык Prolog [8] предполагалось встроить в процессор. Проект не удался, поскольку процессоров требуемой мощности нет до сих пор.

Программа на языке Prolog представляет собой описание:

1. формальной аксиоматической теории (ФАТ) по Д. Гильберту (алфавит, аксиомы, правила вывода);
2. цели (цепочка символов алфавита);
3. максимума используемых ресурсов.

Компьютер стремится вывести цель, не превысив максимума ресурсов. Возможны 2 результата: «Цепочка выводима» или «Не известно, выводима ли цепочка». Язык Prolog относится к логическим, а не процедурным. Его программа не является [8] процедурой - описанием последовательности выполнения представленных команд. Она комбинирует аксиомы с правилами вывода. При этом очень быстро растет граф вывода. В 2000 годах планка требований к ИИ была понижена, для разработки систем ИИ используются процедурные [8] языки программирования.

Неизбежность ошибок в сложных системах также значима для ТОС. Например, сложные программы, включающие миллионы команд, практически не могут не содержать тысяч ошибок. Синтаксические ошибки (несоблюдение формальных правил языка программирования) выявляются автоматически. Основную угрозу несут семантические ошибки (программа не всегда делает то, что предполагалось). Гуманитарные цели и задачи настолько сложны, что их формализация обычно невозможна. Семантические ошибки могут появляться и потому, что человек-разработчик не до конца себе представляет, что он от будущей программы хочет, причем не осознавая этого факта.

5. Примеры распространения ТОС на технику и педагогику

Известный в истории авиации пассажирский и многоцелевой самолет Douglas DC-3 [9] выпускался с 1934 по 1952 годы в США, СССР (Ли-2), Великобритании, Канаде, Австралии, Японии. Всего было построено более 20 тыс. экземпляров. Последняя его версия (C-47) имела: Вес: максимальный взлетный 13200 кг; пустого 7700 кг. Скорость: максимальную 370 км/ч; крейсерскую 300 км/ч. Двигатели поршневые: 2x1350 л.с. Несколько сотен экземпляров эксплуатируются до сих пор. С 1990 года по настоящее время в США производится Basler BT-67 [10]. Он имеет: Вес: максимальный взлетный 13400 кг; пустого 6800 кг. Скорость: максимальную 400 км/ч; крейсерскую 380 км/ч. Двигатели турбо-винтовые: 2x1450 л.с. В конструкции частично используются титан и карбон. Douglas DC-3 модернизируется до Basler BT-67.

Boeing-747 – самый большой в США пассажирский и многоцелевой самолет [11]. Выпускался с 1969 по 2022 годы серийно, далее по настоящее время – по заказам. Кризис. Всего произведено более 1 тыс. экземпляров. Последняя версия (Boeing-747-8) имеет: Вес: максимальный взлетный 443 т; пустого 214 т. Скорость: максимальную $M=0.86$; крейсерскую 990 км/ч. Двигатели турбо-вентиляторные тягой 4x30 т. Очевидно, что это развивающаяся платформа. Вероятно, существуют стратифицированные метамодель и система спецификаций.

В связи с санкциями пассажирская авиация России переживает трудности в производстве самолетов с применением зарубежных технологий. Таков SU-100. Возобновление производства, например, современного аналога турбо-винтового Ил-18 их бы уменьшило. У Ил-18 (по сравнению с SU-100) были выше: дальность и надежность; были ниже: требования к аэродромам, расход топлива, стоимость самолета, крейсерская скорость (650 км/ч против 800, что не критично). Если заменить турбо-винтовые двигатели Ил-18 на менее шумные винто-вентиляторные, то его скорость возрастет. Так было сделано при модернизации транспортного Lockheed C-130 Hercules. Он выпускается с 1954 года по настоящее время. По данным близок к Ан-12. Повышенная шумность старых турбо-винтовых двигателей Ил-18 повлекла бы невозможность использования им многих иностранных аэропортов, но они для российских самолетов закрыты в связи с санкциями.

Диссертации авторов статьи [2, 12] посвящены приложению ТОС к преподаванию соответственно вузовской информатики в целом и информационной безопасности. Были разработаны стратифицированные метамодели и системы спецификаций интерфейсов. Педагогические эксперименты показали рост качества обучения и мотивации студентов.

Авторы статьи предлагали [1] издавать учебники информатики в виде 2 книг – толстой и тонкой. Толстая должна содержать фундаментальную научную информацию, которая, вероятно, быстро не устареет. Тонкая – описывать последние достижения ИТ.

Заключение

Разделами 1–5 соответственно поставленные задачи 1 - 5 были решены, а, следовательно, цель исследования была достигнута.

ЛИТЕРАТУРА

1. Фокин Р.Р., Абиссова М.А. Сервисы обучения информатике и информационным технологиям в высшей школе: Монография. - СПб: изд-во СПбГУСЭ, 2010 – 197с.
2. Фокин Р.Р. Метамодель обучения информатике в высшей школе: автореферат дис. на соиск. уч. степ. докт. пед. наук: 13.00.02 / Рос. пед. ун-т. - СПб, 2000. - 34с.

3. Фокин Р.Р. Некоторые психологические и статистические аспекты преподавания дисциплин из областей математики и информатики в современной высшей школе. // Современные наукоемкие технологии – 2019 - №9 - С.175-179.
4. Знакомьтесь: компьютер. / Под ред. В.М. Курочкина. – М.: Мир, 1989. – 240с.
5. Таненбаум Э. Архитектура компьютера. 5-е изд. – СПб.: Питер, 2007. – 844 с.
6. Фокин Р.Р. Системная оболочка-интерпретатор как инструментальное средство создания АСНТИ в ОС RT-11 // Известия Всесоюзного НИИ гидротехники им. Б.И. Веденеева. – Л.: Гидроатомиздат, 1990. Т.221. – С.15-20.
7. Компьютер обретает разум. / Под ред. В.Л. Стефанюка – М.: Мир, 1990. – 240с.
8. Язык компьютера. / Под ред. В.М. Курочкина. – М.: Мир, 1989. – 240с.
9. Википедия [Электронный ресурс]: Douglas DC-3 // URL: https://ru.wikipedia.org/wiki/Douglas_DC-3 (дата обращения: 26.09.2024)
10. Википедия [Электронный ресурс]: Basler BT-67 // URL: https://ru.wikipedia.org/wiki/Basler_BT-67 (дата обращения: 26.09.2024)
11. Википедия [Электронный ресурс]: Boeing-747 // URL: https://ru.wikipedia.org/wiki/Boeing_747 (дата обращения: 26.09.2024)
12. Абиссова М.А. Сервисы обучения информационной безопасности в теории и методике обучения информатике студентов гуманитарных и социально-экономических специальностей: автореферат дис. на соиск. уч. степ. канд. пед. наук: 13.00.02 / Рос. пед. ун-т. – С-Пб, 2006. – 22с.

© Фокин Роман Романович (vka@mail.ru), Абиссова Марина Алексеевна (marabyss@yandex.ru),
Ершова Елена Викторовна (vka@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»