

Баженова Ирина Васильевна,

старший преподаватель базовой кафедры вычислительных и информационных технологий Сибирского федерального университета, аспирант базовой кафедры информатики и информационных технологий в образовании Красноярского государственного педагогического университета им. В. П. Астафьева; 660041, г. Красноярск, пр. Свободный, 79, e-mail: apkad@ua.ru.

ОСОБЕННОСТИ МЕТОДИКИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ПРОЕКТИВНО-РЕКУРСИВНОЙ СТРАТЕГИИ И КОГНИТИВНЫХ ТЕХНОЛОГИЙ

КЛЮЧЕВЫЕ СЛОВА: проективно-рекурсивная стратегия обучения; результаты обучения программированию; когнитивные подход; ментальная схема; уровни развития алгоритмического и системного мышления.

АННОТАЦИЯ. Преподаватели дисциплины «Информатика и программирование» в высшей школе часто сталкиваются с низким уровнем навыков программирования, алгоритмической грамотности и системного мышления у студентов-первокурсников. В данной статье рассмотрены возможности применения проективно-рекурсивного и когнитивного подходов к обучению программированию и на этой основе предложено обновление целевого, содержательного, технологического и результативно-оценочного компонентов методической системы обучения программированию студентов-математиков. Проективно-рекурсивная стратегия обучения предполагает разработку и использование в учебном процессе самими обучающимися средств обучения – электронных образовательных ресурсов. В процессе обучения программированию целесообразно использовать такие когнитивные средства обучения, как концептуальные карты, понятийные и алгоритмические ментальные карты. Определено понятие «результаты обучения программированию» как совокупность компонентов: 1) уровень развития алгоритмического мышления, 2) уровень развития системного мышления, 3) способность создавать программный продукт на языке программирования. Для каждого компонента выделены определенные индикаторы и критерии оценивания. Приведены результаты педагогического эксперимента по апробации методики в реальном учебном процессе Института математики и фундаментальной информатики Сибирского федерального университета. Эксперимент показал улучшение показателей во всех трех компонентах результатов обучения программированию.

Bazhenova Irina Vasilievna,

Senior Lecturer of Department of Information and Computing Technologies, Siberian Federal University; Post-graduate Student of Department of Informatics and Information Technologies in Education, Krasnoyarsk State Pedagogical University named after V. P. Astafiev, Krasnoyarsk, Russia.

PROJECTIVE-RECURSIVE STRATEGY AND COGNITIVE TECHNOLOGIES IN TEACHING PROGRAMMING TO MATH STUDENTS

KEYWORDS: projective-recursive teaching strategy, results of teaching programming, cognitive approach, mental scheme, levels of development of algorithmic and systematic thinking.

ABSTRACT: Computer science teachers at higher schools are often faced by a low level of programming skills, algorithmic literacy and systematic thinking which many first-year students demonstrate. This article describes the advantages of a projective-recursive and cognitive approach to teaching computer programming. The author proposes the following components of the system of methods of teaching programming be updated: objectives, content, methods, process and assessment. Projective-recursive learning strategy includes the development and use of electronic educational resources designed by the students themselves. Specific cognitive learning tools – concept maps and algorithmic mental maps – may be used in this process. The article introduces the concept of «learning outcomes in programming» as a set of three integrative indicators: the algorithmic thinking level, the systematic thinking level and the ability to design a computer learning program. The results of application of this teaching method in the real learning process at Siberian Federal University demonstrate a significant rise in the level of attainment of each component.

Курс программирования традиционно включается в учебные планы подготовки бакалавров-математиков. Приобретенные в процессе обучения программированию компетенции выступают в роли эффективного инструментального средства для решения задач в их будущей профессиональной деятельности. Важным аспектом обучения программированию является то, что программирование как системная интеллектуальная деятельность существенно влияет на развитие когнитивных способностей обучаемых, в том числе таких спе-

цифических стилей мышления математиков и программистов, как алгоритмическое и системное мышление. Несмотря на значительную методическую проработку теории и практики обучения программированию как в школе, так и в вузе, вопросы совершенствования соответствующих методических систем по-прежнему актуальны. В условиях смены парадигмы высшего образования, реализации новых ФГОС ВПО [11] и профессиональных стандартов [10] к курсу программирования предъявляется требование стать метакогнитивным средст-

вом формирования профессиональных и общекультурных компетенций выпускника вуза. Между тем в существующих методиках обучения программированию студентов-математиков (А. Г. Кушниренко, Г. В. Лебедев, С. С. Лавров, И. А. Кудрявцева, А. Е. Люлькин, С. А. Григорьев, О. В. Смирнова и др.) основное внимание уделено содержательному компоненту методической системы – отбору специфических математико-ориентированных задач [6], [7], [8], [13]. Кроме того, традиционные методики обучения программированию оценивают, главным образом, деятельностный аспект программирования – выполнение лабораторной работы, программную реализацию задачи на экзамене и т. д. Цель данной статьи – предложить, обосновать и доказать эффективность методики обучения программированию студентов-математиков с позиций проективно-рекурсивной стратегии и когнитивного подхода к обучению.

Успех в обучении программированию зависит от многочисленных факторов, обусловленных как объективными, так и субъективными причинами. К числу субъективных факторов следует отнести неготовность

большинства студентов к самостоятельной учебной деятельности, их слабую профессиональную осведомленность, и в силу этого, низкую мотивацию к обучению, особенности «клипового» мышления современного «сетевое» поколения студентов, разнородность студенческой группы в опыте практического программирования. Среди объективных причин, обуславливающих неуспешность освоения курса программирования многими студентами (по данным зарубежных исследователей, до 30% студентов терпят неудачу [6]), наиболее значимой причиной является сложность теории и практики программирования. К сожалению, практическое отсутствие конкурса на математические специальности, обуславливает низкие стартовые показатели первокурсников в обучении программированию. Это подтверждается результатами входного тестирования среди студентов-первокурсников Института математики и фундаментальной информатики Сибирского федерального университета (ИМиФИ СФУ). На рисунке 1 приведены данные тестирования, которые автор проводил в 2012–2014 гг.

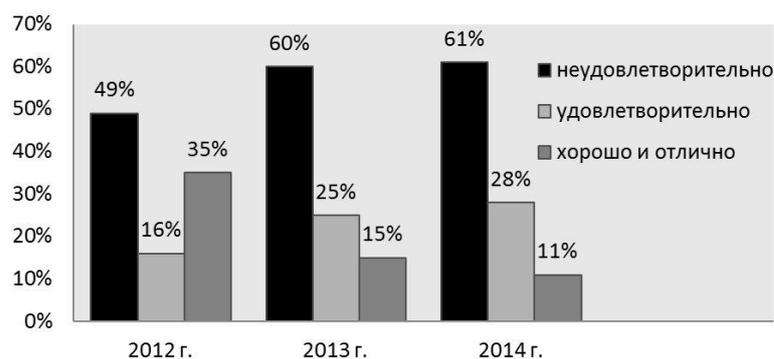


Рис. 1

Гистограмма показывает, что более половины студентов, поступивших на 1 курс, не справляется с заданиями и, к сожалению, имеется тенденция к увеличению этого показателя. Таким образом, комплекс проблем, возникающий в процессе обучения программированию, требует их разрешения (в идеале) или сглаживания при разработке соответствующей методики. Современная эффективная методика обучения программированию студентов-математиков должна не просто помочь студенту овладеть некоторыми практическими навыками программирования, но и способствовать развитию определённых стилей мышления, собственных математикам, умению и желанию учиться самостоятельно и в течение всей жизни. Наше внимание будет обращено к потенциалу проективного и когнитивного подходов к обучению.

Когнитивный подход в обучении сформировался на основе когнитивной науки и связан, в первую очередь, с целенаправленным и последовательным развитием мышления обучаемого на основе его ментального опыта. Научить думать, размышлять, и следовательно, учиться эффективно – сущность когнитивного обучения. Центральным понятием когнитологии является когнитивная (ментальная) схема. Один из основоположников когнитивной психологии У. Найссер когнитивными («предвосхищающими» – у автора) схемами называл обобщенно-визуальные образования, возникающие как результат интеграции зрительных, слуховых и тактильно-осознательных впечатлений [9, с. 71]. Когнитивные схемы имеют сложную иерархическую структуру, образуют системы (сети) схем. От количества, разнообразия, насыщенности

когнитивных схем, прочности связей между ними зависит интеллект личности, успешность в обучении и дальнейшей профессиональной деятельности. Основываясь на этом положении, будем считать, что обучение программированию потребует формирования у обучающихся ментальных схем следующего вида:

– *понятийные ментальные схемы* (включающие схемы-образы и схемы-действия, например, «переменная» – схема-образ и «присваивание» – схема-действие);

– *алгоритмические ментальные схемы*, образующие систему – иерархию схем: от схем 1-го уровня (базовые алгоритмические структуры – следование, ветвление, цикл), до схем – шаблонов программирования.

Наиболее важным представляется формирование схем второго вида, на их основе (ментальном опыте) возможно дальнейшее продвижение обучающегося от программиста-новичка до программиста-профессионала.

Реализовать когнитивный подход к обучению программированию возможно при активном использовании инновационных технологий, к числу которых относят когнитивные технологии, успешно применяемые в сфере образования. Когнитивные технологии обучения – это комплекс методов, приемов, средств, направленных на интеллектуальное развитие обучающихся и учитывающих их индивидуальные ментальные характеристики. Главной целью применения когнитивных технологий в учебном процессе и самообразовании служит существенное улучшение восприятия и понимания, и следовательно, усвоение учебного материала. Важная роль в развитии и активизации когнитивных процессов обучающихся принадлежит средствам визуализации знания, среди которых все большую популярность приобретают концептуальные и ментальные карты.

Концептуальная карта, предложенная Д. Новаком [16], предназначена для систематизации ряда понятий, образующих иерархию и соединенных между собой подписанными стрелками, показывающими характер связи. Для большей эффективности на концептуальную карту помещают примеры, способствующие пониманию отображаемых понятий. Термин «mind map» (ментальная карта) связывается с именем Т. Бьюзена [5], предложившим использовать специальные радиальные схемы, на которых изображены в виде логических связей все вопросы, касающиеся решаемой задачи или изучаемого понятия. Для построения такой карты в большей мере используются ассоциативные связи между включаемыми в карту понятиями. На наш

взгляд, в предметной области программирования целесообразно использовать конкретизированные и частично формализованные ментальные карты в двух направлениях:

– как обучающая *понятийная карта*, на которой представлена в графической форме учебная информация по одному изучаемому понятию;

– как *алгоритмическая ментальная карта*, на которой представлен алгоритм решения задачи, отображающий индивидуальный процесс решения, с возможными тупиковыми ветвями и последующим выбором корректного решения. Такая карта может быть доведена до уровня программной реализации, т. е. стать формальным описанием алгоритма.

Примеры ментальных карт, разработанных для электронного курса программирования, приведены в [2]. Для повышения эффективности обучения необходимо не просто предложить готовые карты студентам для изучения, а привлечь их к созданию таких карт. В этом случае они становятся не только средствами визуализации знания, но и программными продуктами, созданными самими студентами, которые можно использовать для оценивания результатов обучения программированию. При разработке концептуальных и ментальных карт возможно применить проективно-рекурсивную стратегию, предложенную в [12], где вводится понятие проективной методической системы как открытой системы, моделируемой и развивающейся в виде проекта с определенными характеристиками и поведением в настоящем и будущем. Инициирование, моделирование и реализация проективной методической системы обучения предполагает проектирование и динамичное развитие всех компонентов системы с привлечением всех участников проекта: студентов, преподавателей, заказчиков (например, возможных работодателей или студентов и преподавателей других вузов). Ряд вопросов, связанных с разработкой методической системы обучения программированию, обладающих указанными свойствами, рассмотрены автором в [1]. Суть проективно-рекурсивной стратегии в обучении – в реализации двух базовых принципов. Принцип проективности заключается в том, что будущая профессиональная деятельность обучающихся проектируется на настоящий учебный процесс, в ходе которого моделируются условия для решения возможных профессиональных задач. Принцип рекурсивности предполагает, что в процессе обучения обучающиеся создают и используют электронные учебные ресурсы. Рассмотрим возможности применения проективно-рекурсивной стра-

тегии к обновлению всех компонентов методической системы обучения программированию. В структуре методической системы выделим следующие компоненты: целевой, содержательный, технологический (включающий методы, средства и формы обучения), результативно-оценочный (включающий результаты, средства диагностики и контроля обучения).

В *целевом компоненте* сформулируем главную цель разработки методической системы – достижение результатов обучения, которые можно представить на 3 уровнях: *дисциплинарном* – формирование специфических компетенций в области программирования; *профессиональном* – формирование профессиональных компетенций бакалавра-математика; *метадисциплинарном* – развитие алгоритмического и системного мышления, формирование общекультурных и общепрофессиональных компетенций. Развитие алгоритмического и системного мышления непосредственно влияет на формирование ряда общекультурных, общепрофессиональных и профессиональных компетенций. В то же время взаимосвязано развитие компетенций в области программирования и алгоритмического и системного мышления, представляя собой систему вложенных (рекурсивных) целей и одновременно результатов обучения.

Проектируя *содержательный компонент* методической системы как проекцию целей и результатов обучения, приходим к выводу, что следует отойти от традиционного решения типовых задач по программированию. Целесообразна работа в следующих направлениях:

- решение профессионально ориентированных задач, причем с элементами опережающего обучения, т. е. задач, не только из предметных областей параллельно изучаемых математических дисциплин, но и изучаемых на втором-третьем курсах;

- создание студентами программ-тестов на знание синтаксических, семантических и прагматических особенностей языка программирования, при этом тесты кодируются на этом языке (пример рекурсивности). Важно, что самостоятельное формулирование тестовых вопросов способствует пониманию и усвоению пройденного учебного материала;

- вовлечение студентов в практическую работу над проектами межвузовской кооперации, реализующими сотрудничество студентов и преподавателей разнопрофильных вузов [3] и проектами IT-бизнеса, дающими возможность студентам сделать первые шаги в профессиональной сфере.

Отметим следующий факт: программы-тесты, создаваемые студентами, являются

не только элементами содержательного компонента методической системы, но выступают в роли электронных средств обучения (*технологический компонент*), которыми могут воспользоваться не только сокурсники, но в будущем следующее поколение студентов. Подобная практика формирует положительную мотивацию к овладению знаниями и умениями в программировании у студентов и повышает их ответственность за собственные результаты обучения.

Обратимся к *результативно-оценочному компоненту* методической системы. В официальных документах Болонского процесса под результатами обучения понимаются наборы компетенций, выражающих, что именно студент будет знать, понимать и способен делать после завершения процесса обучения [4]. Придерживаясь этого определения, сформулируем утверждение:

Результаты обучения программированию должны быть выражены в достижении следующих интегративных показателей: 1) уровня развития алгоритмического мышления; 2) уровня развития системного мышления; 3) способности создавать программные продукты на изучаемом языке программирования на основе приобретенных знаний, понимания и умений.

Данные показатели являются взаимопроницаемыми, носят интегративный характер и могут быть представлены комплексом индикаторов и соответствующими критериями оценивания.

Индикаторы уровня развития алгоритмического мышления:

- 1) способность анализировать и специфицировать множество входных и выходных данных задачи;

- 2) способность представить решение задачи через элементарные действия, включающие базовые алгоритмические структуры;

- 3) способность выполнить функциональную декомпозицию задачи;

- 4) способность анализировать и учитывать различные варианты (сценарии) выполнения алгоритма;

- 5) способность оценить эффективность алгоритма;

- 6) способность оптимизировать, улучшить алгоритм.

При этом соответствующие *критерии оценивания* определены как а) начальный: наличие индикаторов 1), 2); б) средний: наличие индикаторов 1) – 4); в) высокий: наличие индикаторов 1) – 6).

За *индикаторы уровня развития системного мышления* можно принять следующие характеристики: 1) способность анализировать и моделировать предметную область задачи; 2) способность эффективно

организовать данные; 3) способность структурировать алгоритм и программное решение задачи; 4) способность выделить факторы, влияющие на состояние отдельного объекта и всей системы; 5) способность адаптировать решение к другой проблемной области; 6) способность интегрировать разработанный алгоритм и программное решение с другими разработками; 7) способность объяснить алгоритм и программное решение другим людям на соответствующем уровне; 8) способность к интеграции математического знания и теории и практики программирования.

Для оценивания уровня развития системного мышления целесообразно использовать таксономию результатов обучения SOLO (Структура Наблюдаемых Результатов Обучения), предложенную J. V. Biggs и K. F. Collis [15]. Согласно этой таксономии, существуют 5 качественно различных уровней мышления: доструктурный, одноструктурный, многоструктурный, реляционный, расширительно-абстрактный. Первые три уровня могут характеризовать репродуктивное знание с различной степенью усвоения, четвертый уровень показывает понимание изучаемого предмета, пятый уровень – уровень продуктивного знания и творчества. Данная таксономия может быть применена к построению студентами концептуальных карт.

Способность создавать программный продукт (третий показатель результатов обучения) предлагается выразить следующими индикаторами и критериями оценивания:

1) студент способен разработать и отладить (устранить синтаксические ошибки и ошибки времени выполнения) простую программу с базовыми алгоритмическими структурами и типами данных – *пороговый уровень*;

2) студент способен разработать, отладить и тестировать программный код для типовых заданий на основе стандартных алгоритмов и имеющихся программных решений – *базовый уровень*;

3) студент умеет работать с абстрактными типами данных, самостоятельно разрабатывает алгоритмы и программно реализует их – *высокий уровень*.

Все вышесказанное было положено в основу методики обучения программированию, апробированной в Институте математики и фундаментальной информатики СФУ в 2012–2014 годах. В педагогическом эксперименте участвовали 124 студента 1-го курса направлений подготовки 01.03.01 «Математика» и 01.03.02 «Прикладная математика и информатика». Результаты студентов экспериментальной группы были соотнесены с результатами студентов контрольной группы и отображены на рис. 2.

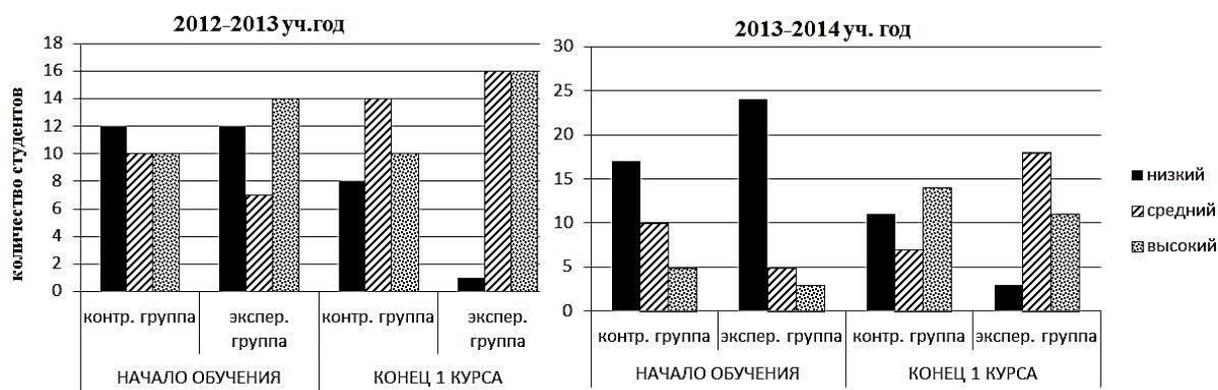


Рис. 2. Результаты педагогического эксперимента

Как видно из гистограммы, в экспериментальных группах наблюдается значительное сокращение числа студентов, показывающих низкие результаты обучения, по сравнению с контрольными группами, что свидетельствует об эффективности применяемой методики.

Таким образом, использование проективно-рекурсивной стратегии для обновления компонентов методической системы обучения программированию позволяет достичь планируемых результатов обучения

студентов в виде необходимого уровня развития алгоритмического и системного мышления и способности создавать программные продукты. Сочетание в учебном процессе традиционных технологий обучения с инновационными, опирающимися на когнитивный подход, дает дидактический эффект в виде наличия у студентов высокой мотивированности, заинтересованности, самостоятельности и ответственности за результаты учебной и исследовательской деятельности.

Л И Т Е Р А Т У Р А

1. Баженова И. В. Проектирование содержания и результатов обучения программированию студентов-математиков // Вестник КГПУ им. В. П. Астафьева. 2013. № 2(24), с.79–82.
2. Баженова И. В. К вопросу отбора содержания курса «Информатика и программирование» для студентов математических направлений подготовки на основе проективно-рекурсивной стратегии // Материалы междунауч.-практич. конференции ФИИТМ-2014/ Красноярск : СФУ, 2014, С. 18–23.
3. Баженова И. В., Гринберг Г. М., Ивкина Л. М. Развитие компетенций будущих педагогов и инженеров в условиях межвузовской кооперации // Вестник Челябинского государственного педагогического университета. 2014. № 2. С. 62–69.
4. Болонский процесс: Результаты обучения и компетентностный подход (книга-приложение 1) / под науч. ред. В. И. Байденко. М. : Исследовательский центр проблем качества подготовки специалистов, 2009.
5. Бьюзен Т. Научите себя думать : пер. с англ. 2-е изд. Минск : Попурри, 2004.
6. Кудрявцева И. А. Методика обучения бакалавров физико-математического образования математическим основам парадигм программирования : дис. ... канд. пед. наук. СПб., 2005.
7. Кушниренко А. Г., Лебедев А. Г. Программирование для математиков. М. : Наука, 1988.
8. Лавров С. С. Программирование. Математические основы, средства, теория : учебное пособие. СПб. : БХВ-Петербург, 2001.
9. Найссер У. Познание и реальность. М. : Прогресс, 1981.
10. Об утверждении профессионального стандарта «Программист»: [приказ Мин. труда и соц. защиты РФ от 18.11.2013 г. № 679н]. URL: <http://www.garant.ru/products/ipo/prime/doc/70447858>.
11. Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.01 Математика (уровень бакалавриата): [приказ Минобрнауки России от 07.08.2014 N 943]. URL: http://www.consultant.ru/document/cons_doc_LAW_168629.
12. Пак Н. И. Проективный подход в обучении как информационный процесс : монография. Красноярск : РИО КГПУ, 2008.
13. Смирнова О. В. Разработка содержания обучения программированию в системе информационной подготовки бакалавра математики : дис. ... канд. пед. наук. М., 2005
14. Bennedsen, J., Caspersen, M.E. Failure rates in introductory programming / SIGCSE Bull. 39(2), 2007, pp. 32–36.
15. Biggs J.B., Collis K.F. Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome). –New York: Academic Press, 1982.
16. Novak, J.D., Gowin, D.B. Learning How to Learn. New York: Cambridge University Press, 1984.

L I T E R A T U R E

1. Bazhenova I. V. Proektirovanie soderzhaniya i rezul'tatov obucheniya programmirovaniyu studentov-matematikov // Vestnik KGPU im. V. P. Astaf'eva. 2013. № 2(24), s.79–82.
2. Bazhenova I. V. K voprosu otbora soderzhaniya kursa «Informatika i programmirovanie» dlya studentov matematicheskikh napravleniy podgotovki na osnove proektivno-rekursivnoy strategii // Materialy mezhd. nauch.-praktich. konferentsii FIITM-2014/ Krasnoyarsk : SFU, 2014, S. 18–23.
3. Bazhenova I. V., Grinberg G. M., Ivkina L. M. Razvitie kompetentsiy budushchikh pedagogov i inzhenerov v usloviyakh mezhvuzovskoy kooperatsii // Vestnik Chelyabinskogo gosudarstvennogo pedagogicheskogo universiteta. 2014. № 2. S. 62–69.
4. Bolonskiy protsess: Rezul'taty obucheniya i kompetentnostnyy podkhod (kniga-prilozhenie 1) / pod nauch. red. V. I. Baydenko. M. : Issledovatel'skiy tsentr problem kachestva podgotovki spetsialistov, 2009.
5. B'yuzen T. Nauchite sebya dumat' : per. s angl. 2-e izd. Minsk : Popurri, 2004.
6. Kudryavtseva I. A. Metodika obucheniya bakalavrov fiziko-matematicheskogo obrazovaniya matematicheskim osnovaniyam paradigim programmirovaniya : dis. ... kand. ped. nauk. SPb., 2005.
7. Kushnirenko A. G., Lebedev A. G. Programmirovanie dlya matematikov. M. : Nauka, 1988.
8. Lavrov S. S. Programmirovanie. Matematicheskie osnovy, sredstva, teoriya : uchebnoe posobie. SPb. : BKhV-Peterburg, 2001.
9. Naysser U. Poznanie i real'nost'. M. : Progress, 1981.
10. Ob utverzhdenii professional'nogo standarta «Programmist»: [prikaz Min. truda i sots. zashchity RF ot 18.11.2013 g. № 679n]. URL: <http://www.garant.ru/products/ipo/prime/doc/70447858>.
11. Ob utverzhdenii federal'nogo gosudarstvennogo obrazovatel'nogo standarta vysshego obrazovaniya po napravleniyu podgotovki 01.03.01 Matematika (uroven' bakalavriata): [prikaz Minobrnauki Rossii ot 07.08.2014 N 943]. URL: http://www.consultant.ru/document/cons_doc_LAW_168629.
12. Pak N. I. Proektivnyy podkhod v obuchenii kak informatsionnyy protsess : monografiya. Krasnoyarsk : RIO KGPU, 2008.
13. Smirnova O. V. Razrabotka soderzhaniya obucheniya programmirovaniyu v sisteme informatsionnoy podgotovki bakalavra matematiki : dis. ... kand. ped. nauk. M., 2005
14. Bennedsen, J., Caspersen, M.E. Failure rates in introductory programming / SIGCSE Bull. 39(2), 2007, pp. 32–36.
15. Biggs J.B., Collis K.F. Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome). –New York: Academic Press, 1982.
16. Novak, J.D., Gowin, D.B. Learning How to Learn. New York: Cambridge University Press, 1984.

Статью рекомендует кандидат тех. наук, д-р пед. наук, доцент М. В. Лапенюк.