Э. А. Нигматулина, Н. И. Пак,

Красноярский государственный педагогический университет им. В. П. Астафьева

СТУДЕНТ-ЦЕНТРИРОВАННОЕ ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ В ПЕДАГОГИЧЕСКОМ ВУЗЕ

Аннотация

Статья посвящена обоснованию педагогической идеи реализации студент-центрированного обучения в педагогических вузах на примере предметной подготовки по программированию. Обоснована необходимость создания и развития новой ментальной учебной платформы, обеспечивающей реализацию инновационных моделей индивидуализации и дифференциации обучения программированию и развития когнитивных способностей обучаемых, в частности алгоритмического стиля мышления. Показана роль информатизации учебного процесса в повышении качества подготовки студентов по программированию.

Ключевые слова: студент-центрированное обучение, ментальная учебная платформа, полиязыковое обучение программированию, программистский стиль мышления.

Введение

В настоящее время во многих странах все чаще стали обращать внимание на парадигму студент-центрированного обучения [12–14]. Студент-центрированное обучение — это система, нацеленная на непринужденное образование и создание условий, обеспечивающих мотивацию к обучению, развитие личности обучаемого, гуманное отношение к нему. Она требует от студента быть активным и ответственным участником в построении собственной образовательной траектории, в выборе темпа обучения, средств и способов достижения образовательных результатов.

Почему студент-центрированное обучение становится привлекательным в настоящее время?

Делая прогнозы на будущие профессиональные компетенции, которыми должен обладать специалист, ученые связывают их с когнитивной деятельностью в высокотехнологичных производственной и бизнес-сферах, с высоким уровнем алгоритмического мышления [2].

Научно-технический прогресс и скорость смены технологий заставляют максимально плотно сблизить теоретическую подготовку студентов с производством и наукой. Обучаясь в вузе, студенты уже сами осознают необходимость приобретения реальных профессиональных компетенций для успешности в своей будущей профессиональной деятельности.

Однако традиционные образовательные модели не позволяют естественным образом реализовать принципы студент-центрированного обучения в предметной подготовке. Попытки «вседозволенности» и демократизации студенческого самоуправления носят искусственный характер, часто требуют серьезной перестройки образовательного процесса в вузе, что достаточно сложно осуществить в реалиях сложившихся укладов и отношений в высшей школе.

Целью статьи является обоснование необходимости создания новой ментальной учебной платформы, обеспечивающей реализацию инновационных моделей индивидуализации и дифференциации обучения программированию студентов педагогических вузов на принципах студент-центрированной парадигмы образования.

Контактная информация

Нигматулина Эльмира Альфредовна, аспирант Красноярского государственного педагогического университета им. В. П. Астафьева; *адрес:* 660049, г. Красноярск, ул. Перенсона, д. 7; *телефон:* (391) 263-97-34; *e-mail:* nigmira@yandex.ru

Пак Николай Инсебович, доктор пед. наук, профессор, зав. базовой кафедрой информатики и информационных технологий в образовании Красноярского государственного педагогического университета им. В. П. Астафьева; *адрес:* 660049, г. Красноярск, ул. Перенсона, д. 7; *телефон:* (391) 263-97-33; *e-mail:* nik@kspu.ru

E. A. Nigmatulina, N. I. Pak,

Krasnoyarsk State Pedagogical University named after V. P. Astafyev

STUDENT-CENTERED TRAINING IN PROGRAMMING IN A PEDAGOGICAL UNIVERSITY

Abstract

The article is devoted to the justification of pedagogical idea of implementing a student-centered teaching in pedagogical universities on an example of subject training in programming. The necessity to create and develop a new mental training platform that provides implementation of innovative models of individualisation and differentiation of training in programming and the development of cognitive abilities of learners, particularly their algorithmic style of thinking is substantiated. The role of informatization of educational process in improving the quality of training students in programming is shown.

Keywords: student-centered teaching, mental training platform, multilingual training in programming, programmer's way of thinking.

Методологическая часть

Современный мир носит характер информационного и умного общества. Усугубляется противоречие между его требованиями к высшему образованию и недопустимо отстающими в этом образовании методами и средствами обучения.

Моделирование новых образовательных моделей и технологий сегодня необходимо осуществлять с позиций когнитивного подхода и образовательных технологических платформ в условиях глобализации образовательного процесса, электронных форм и средств обучения [10].

Достигнутый уровень методической науки в высшей школе позволяет создавать и развивать *образовательные технологические платформы*, которые обеспечивают построение инновационного учебного процесса образовательных учреждений всех уровней адекватно вызовам современного общества [8].

Для организации студент-центрированной предметной подготовки студентов целесообразно сформировать *новую ментальную учебную платформу*, которая позволяет обеспечить возможности непринужденного обучения без существенной перестройки регламентов традиционного учебного процесса в условиях его информатизации.

Учебная платформа — это структурно-концептуальная методологическая основа для создания методических систем обучения предмету.

Традиционная предметная учебная платформа (рис. 1) имеет знаниевый характер, определяет механизм формирования когнитивных способностей обучаемого (компетенций) путем создания методической системы его предметной подготовки (обучение теории и практике) согласно принципам современной дидактики. Эта платформа обеспечивает развитие когнитивных способностей ученика с помощью содержательных возможностей дисциплины по модели «черного ящика».

В настоящее время, используя результаты моделирования познавательных способностей человеческого мозга, когнитивные технологии обучения (комплекс методов, приемов, средств, учитывающих индивидуальные ментальные характеристики обучающихся), удается создавать методы прямого или косвенного интеллектуального развития обучающихся, достигать лучшего понимания учебного материала. В этой связи появляется возможность определить новую когнитивную (ментальную) учебную платформу предметного обучения. В отличие от традиционной, в ней определяются способы развития необходимых когнитивных способностей ученика по модели «белого ящика». Затем содержание и процессуальные компоненты дисциплины подстраивают под эти механизмы. Другими словами, методическая система предметного обучения видоизменяется в части целей обучения и последовательности учебных действий, определяя доминирующую стратегию — от когнитивной практики к теории (рис. 2).

Мышление является функцией мозга и представляет собой естественный непрерывный информационный процесс. В основе механизма мышления лежат ментальные схемы в виде зафиксированных в памяти ощущений и метаощущений окружающего мира в пространстве и во времени, обогащенные модельными и понятийными категориями [3, 9].

Ментальная схема является динамичной, меняется со временем эволюционно путем добавления новых образов, реконструкции старых в пространстве и во времени. Совокупность ментальных схем образует структуру тезауруса человека. Простейшая ментальная схема как целостная структура состоит из элементарных образов объектов и событий.

Изучение любой предметной области должно протекать по этим же правилам: сначала необходимо обеспечить формирование чувственных ментальных схем предметной области с помощью их ментальных

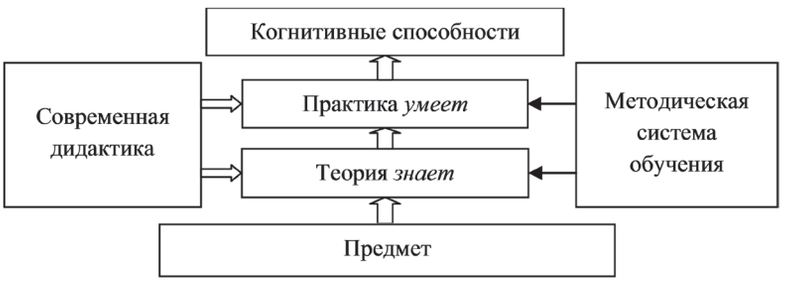

Рис. 1. Традиционная предметная учебная платформа

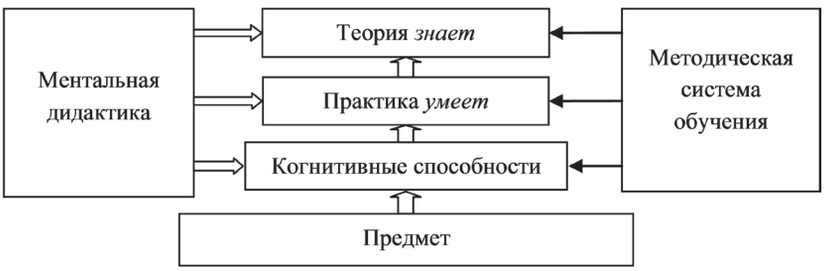

Рис. 2. Ментальная учебная платформа

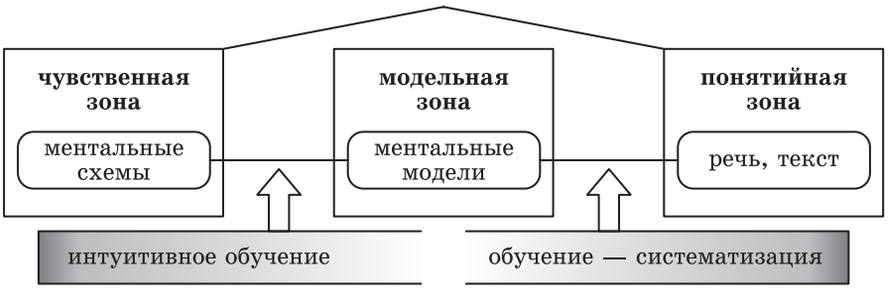

Рис. 3. Основания ментальной дидактики

моделей, затем систематизировать все образы, модели и понятия средствами языка (рис. 3).

Следовательно, сущность обучения сводится к двум этапам. На первом этапе обучение нацелено на *формирование связей между образами чувственной зоны и их модельными представлениями*. Это этап формирования чувственного разума — *интуиции*. Второй этап обучения назовем *систематизацией интуитивного опыта с помощью понятий и терминов* предметной области.

На этапе интуитивного обучения целесообразно учебный материал представлять в виде ментальных схем и интеллект-карт [7], а на этапе систематизации важно закрепить (регулярно активировать ментальные схемы предыдущего этапа обучения) с помощью решения частных задач и информационного описания (теории) знаний предметной области.

Методические системы обучения, реализующие вышеназванные положения ментальной дидактики, нацелены в первую очередь на развитие мышления с помощью теории, средств и методов предметных областей.

Практическая часть

В качестве примера организации студент-центрированного обучения на основе ментальной учебной платформы рассмотрим особенности формирования программистского стиля мышления у будущих учителей информатики в процессе обучения по курсу «Языки и методы программирования».

Термин «программистский стиль мышления» (а этот стиль эмпирически наблюдался психологами, которые исследовали поведение людей, связанных с вычислительными машинами) отражает значительную роль программистов в формулировке и решении важнейшей социальной задачи — формировании нового поколения людей, способных активно жить в условиях современного информационного общества [11].

Существует явное заблуждение, связанное с мыслью о том, что хорошая математическая подготовка и логическое мышление человека являются необходимыми и достаточными условиями для легкого приобретения им компетенций программиста. Математические способы решения задач в виде математических алгоритмов представляют ментальные схемы, *не соответствующие* алгоритмическим конструкциям решения задач на компьютере. У человека с явно выраженным математическим складом мышления возникает непонимание и неприятие машинно-ориентированных алгоритмов. Многие практикующие учителя давно заметили этот синдром сильного математика, неспособного освоить азы программирования.

В этой связи возникает психолого-педагогическая проблема обучения программированию студентов с разными уровнями математического мышления. Особую актуальность приобретает данная проблема для подготовки будущих учителей информатики, поскольку им, в дополнение ко всему, предстоит овладеть методами формирования программистского стиля мышления у школьников.

А. П. Ершов в своей статье «О человеческом и эстетическом факторах программирования» писал: «Программист обязан обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдисоновским талантом сооружать все что угодно из нуля и единицы, он должен соединять в себе аккуратность бухгалтера с проницательностью разведчика, фантазию автора детективных романов с трезвой практичностью бизнесмена, а кроме того, иметь вкус к коллективному труду, быть лояльным к организатору работ и так далее… Программист — солдат второй промышленной революции и как таковой должен обладать революционным мышлением и мужеством» [цит. по 11].

Проблемам обучения курсу программирования школьников и студентов посвящено много исследований, среди которых можно назвать работы И. В. Баженовой, В. В. Бобкова, А. И. Газейкиной, А. П. Ершова, В. Е. Жужжалова, Г. А. Звенигородского, В. В. Калитиной, А. А. Кузнецова, Е. К. Хеннера и многих других. Однако в них не уделено должного внимания особенностям формирования программистского стиля мышления (ПСМ).

Для достижения высокого уровня сформированности ПСМ студента, согласно ментальной учебной платформе и принципам студент-центрированного обучения, необходимо выделить следующие факторы:

- для оценки уровня ПСМ студента использовать многокомпонентную модель машинно-ориентированного алгоритмического мышления;
- для формирования у студентов с низким уровнем алгоритмического мышления понимания базовых конструкций машинно-ориентированных алгоритмов применять средства и методы телесного подхода;
- для переориентации студентов с высоким уровнем математического алгоритмического мышления на машинно-зависимый алгоритми-

ческий стиль использовать натурные средства алгоритмических конструкций;
- для учета личностных предпочтений студента использовать полиязыковый метод изучения различных парадигм программирования;
- для самообразовательной деятельности студентов в сфере алгоритмизации и программирования создать специальную личностно-ориентированную информационно-образовательную предметную среду [4].

Остановимся более подробно на этих моментах.

Модель оценки уровня программистского стиля мышления студента

На современном этапе развития информатики для успешного взаимодействия с компьютером необходим стиль мышления, который можно назвать *объектным*. Он предполагает умение разделить сложную систему на объекты и выстроить их иерархию, т. е. произвести объектную декомпозицию системы, а затем описать поведение этих объектов.

Процесс разработки высокоинтеллектуальной экспертной программной системы существенно упрощается при использовании *логической* парадигмы. Изучение логической парадигмы обработки информации способствует развитию у студентов другого стиля мышления, предполагающего отказ от императивных стереотипов.

Идея *функционального* программирования опирается на интуитивное понятие о функциях как о достаточно общем механизме представления и анализа решений сложных задач.

Технология *параллельного программирования* не выделяется в отдельную парадигму, так как она может быть реализована в каждой из них. В отличие от программирования последовательных вычислений, концептуальную основу которого составляет понятие алгоритма, реализуемого по шагам строго последовательно во времени, в параллельном программировании программа порождает совокупность параллельно протекающих процессов обработки информации, полностью независимых или связанных между собой статическими или динамическими пространственно-временными, причинно-следственными или информационными отношениями. В процессе изучения параллельной технологии программирования формируется *параллельный стиль мышления*, предполагающий способность к предварительному умозрительному «распараллеливанию» поставленной задачи. При достаточно высоком уровне развития параллельного стиля мышления программист способен предвидеть те проблемы, возникающие при работе параллельных алгоритмов, которые не встречаются в работе последовательных [6].

Таким образом, вузовский курс программирования требует значительного расширения трактовки алгоритмического стиля мышления. Чтобы не путать с его исторически сложившимся традиционным пониманием, именно этот стиль мышления современного программиста мы будем называть «*программистским стилем мышления*», многокомпонентная модель которого представлена на рисунке 4.

Оценку уровня сформированности ПСМ студента следует проводить в рамках интегральной порядковой шкалы по совокупности указанных в модели компонентов. В модель ПСМ будущих учителей информатики включен методический компонент. Педагог должен обладать не только высоким уровнем алгоритмического мышления, но и способностью формировать и развивать алгоритмическое мышление своих учеников.

Средства и методы телесного подхода

Основные положения теории телесного подхода [1] дают основание использовать кинестетические тренажеры для формирования чувственных образов алгоритмических конструкций на интуитивном уровне.

Например, для моделирования понятия переменной можно использовать специальные ящики. Имя переменной — это буква, записанная на ящике, а присваиваемое ей значение — это величина (число), помещаемая в ящик. Манипуляции с подобными ящиками при словесных записях в виде алгоритмических инструкций формируют ментальные схемы на чувственном уровне. Составление программы на языке программирования по данному алгоритму интерпретируется как перевод с языка обычного на язык программирования. При этом ящики для переменных заменяются на ячейки памяти ЭВМ.

Большие «телесные» алгоритмические возможности имеет конструктор LEGO, в котором фигуры и конструкции собираются из различных частей-кубиков. Инструкция по сборке конкретной конструкции легко понимается как алгоритм на интуитивном уровне.

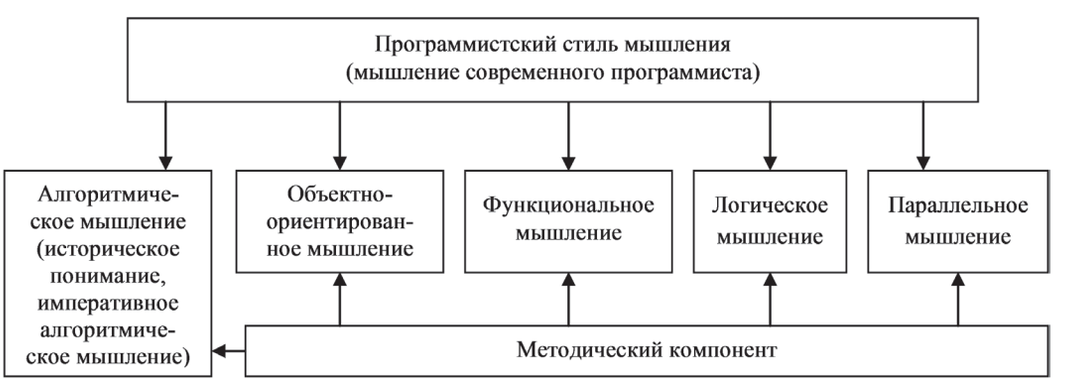

Рис. 4. Многокомпонентная модель программистского стиля мышления

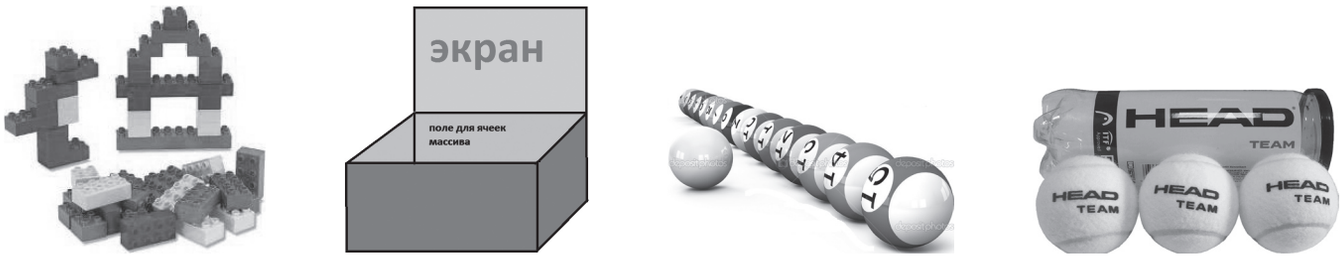

Рис. 5. Кинестетические тренажеры

На рисунке 5 показаны примеры некоторых кинестетических тренажеров, которые следует применять на этапе интуитивного обучения.

Натурные средства алгоритмических конструкций

Обучение с использованием натурных средств существенно облегчает понимание сложных фундаментальных понятий алгоритмизации и программирования. При этом с точки зрения подготовки будущего учителя информатики интерес вызывает самостоятельная разработка и использование натурных средств студентами. В этом случае обучение будет осуществляться в действии и через действия. Студенты сами создают натурные средства обучения и используют их в своей подготовке и в обучении школьников курсу информатики.

Так, например, при изучении числовых алгоритмов по теме «Системы счисления» можно использовать счеты (рис. 6), с помощью которых сначала разбираются базовые понятия счета и количества, визуализируются процессы позиционирования цифр числа, «вручную» разбирается алгоритм перехода из одной системы счисления в другую, а уже затем строится алгоритм или программный код. Счеты представляют собой доску с ячейками, в которые поочередно раскладываются костяшки, начиная с нижней правой ячейки и вверх. Для указания системы счисления, в которой работает обучаемый, используется ползунок. В случае заполнения столбца-разряда, т. е. попадания в ячейку ползунка, все костяшки из столбца забираются и заменяются одной костяшкой, которая кладется в старший разряд. Так продолжается до тех пор, пока не закончится предложенная куча костяшек. Данный процесс позволяет продемонстрировать переход в следующий разряд числа и объясняет, что каждая единица старшего разряда представляет n костяшек предыдущего (n — основание системы счисления).

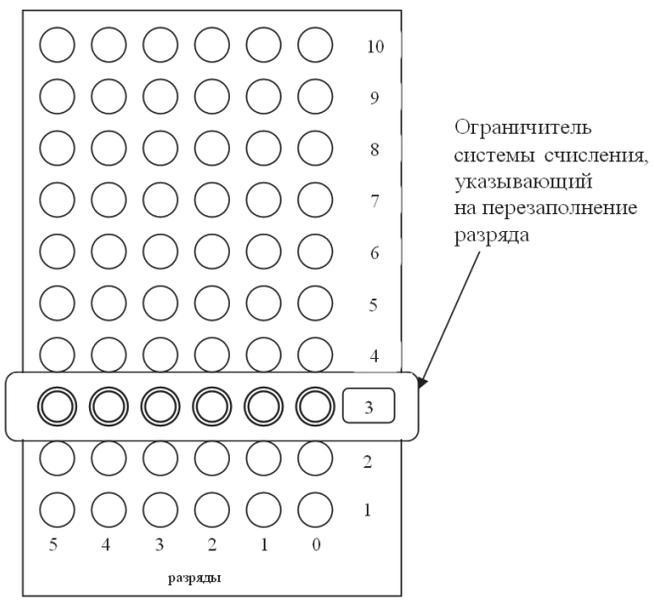

Рис. 6. Счеты в различных системах счисления

Натурные средства обучения существенным образом могут исправить наличие критичного дефицита в механизме обучения информатике, ликвидировав саму причину появления этого дефицита, упорядочив получаемую систему знаний и восстановив дидактически верную структуру перехода от сенсорного оперирования к абстрактному.

Полиязыковый метод изучения различных парадигм программирования

В настоящее время выделяют четыре парадигмы: процедурную, объектно-ориентированную, логическую и функциональную. Во многих образовательных учреждениях, как правило, обучают лишь одному из подходов, а будущие учителя информатики владеют одним, в лучшем случае, двумя языками программирования, которые обычно относятся к одной из существующих парадигм. Учитель информатики использует тот подход в программировании, которому его научили [5].

Практически во всех исследованиях по рассматриваемой теме предлагается изучать различные парадигмы последовательно, и исследователи спорят лишь об оптимальной последовательности их изучения. При проектировании методической системы обучения курсу «Языки и методы программирования» в рамках ментальной учебной платформы целесообразно предусмотреть *параллельное изучение парадигм программирования* с помощью системы практико-ориентированных задач. Студентам предлагается решить конкретную задачу средствами языков программирования, относящихся к различным парадигмам.

Подобный *полиязыковый подход* позволяет студенту критически анализировать стили и парадигмы программирования.

Пример вычисления факториала представлен на рисунке 7.

Пример ввода данных для решения системы линейных уравнений методом прогонки на языках Pascal и Delphi показан на рисунке 8.

Процедурная парадигма (Pascal)	Логическая парадигма (Prolog)

program XXX;
var F, i: integer;
begin
i:=0;
F:=1;
writeln('Введите n');
readln (n);
while i<n do
begin
Inc(i);
F:=F*i;
end;
writeln(' ', n, '=, F);
end.
	Predicates
	Fact(integer, real)
	clauses
	Fact(0, 1):-!.
	Fact(X, FactX):-Y=X-1,
	Fact(Y, FactY),
	FactX=X Fact Y.

Рис. 7. Пример вычисления факториала

Процедурная парадигма (Pascal)	Объектно-ориентированная парадигма (Delphi)

writeln('Vvedite elementi glavnoi diagonali: ');
for i:=1 to 4 do
begin
read(b[i]);
end;
 	
writeln('Vvedite elementi a: ');
for i:=2 to 4 do
begin
read(a[i]);
end;
 	
writeln('Vvedite elementi c: ');
for i:=1 to 3 do
begin
read(c[i]);
end;
 	
writeln('Vvedite rezyltati: ');
for i:=1 to 4 do
begin
read(f[i]);
end;
	procedure TForm1.Button1Click(Sender: TObject);
	begin
	{вводим элементы главной диагонали}
	b[1]:=strtoint(edit1.Text);
	b[2]:=strtoint(edit2.Text);
	b[3]:=strtoint(edit3.Text);
	b[4]:=strtoint(edit4.Text);
	{вводим элементы, стоящие под главной диагональю}
	a[2]:=strtoint(edit5.Text);
	a[3]:=strtoint(edit6.Text);
	a[4]:=strtoint(edit7.Text);
	{вводим элементы, стоящие над главной диагональю}
	c[1]:=strtoint(edit8.Text);
	c[2]:=strtoint(edit9.Text);
	c[3]:=strtoint(edit10.Text);
	{вводим результаты}
	f[1]:=strtoint(edit11.Text);
	f[2]:=strtoint(edit12.Text);
	f[3]:=strtoint(edit13.Text);
	f[4]:=strtoint(edit14.Text);

Рис. 8. Пример ввода данных для решения системы линейных уравнений методом прогонки

Схема организации учебного процесса полиязыковым методом представлена на рисунке 9.

Безусловно, полиязыковый метод требует больших временны́х затрат и умственных усилий студента, но он по сути реализует студент-центрированный принцип обучения. Разные стили мышления студентов позволяют им индивидуально уловить смысл одного из способов представления алгоритма задачи, а затем с легкостью освоить другие парадигмы программирования.

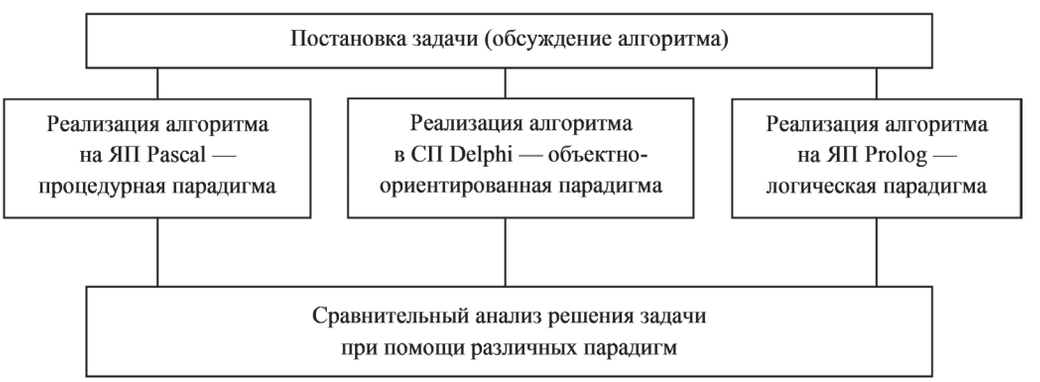

Рис. 9. Структурная схема полиязыкового решения задачи

Самообразовательная деятельность студента в информационно-образовательной предметной среде

Для организации студент-центрированной самостоятельной деятельности студента по курсу «Языки и методы программирования» удобно использовать информационно-образовательную среду Moodle, в которую включены следующие учебные материалы:

- интерактивный электронный курс;
- визуализированные источники теоретического материала;
- практические и лабораторные работы, содержащие комплексы разноуровневых и практико-ориентированных заданий;
- тестовые задания;
- модульно-рейтинговая система контроля успеваемости;
- необходимый справочный материал;
- тренажеры;
- электронные учебники и пособия.

Многообразие методического обеспечения курса создает необходимые условия для повышения познавательной активности обучаемого и реализации студент-центрированного обучения в процессе самообразования студента. Использование системы Moodle в образовательном процессе позволяет формировать у обучающихся способность к самостоятельному поиску, к постоянному, непрерывному самообразованию, стремление к творческому использованию знаний на практике, что обеспечивает более высокое качество освоения содержания учебной дисциплины при более рациональном использовании времени обучающегося.

Выводы

Реализация студент-центрированного предметного обучения в педагогических вузах возможна при условии использования новой ментальной учебной платформы, обеспечивающей создание методических систем развития необходимых когнитивных способностей обучающегося по модели «белого ящика».

На примере предметной подготовки по программированию студентов педагогического вуза обоснована эффективность ментальной учебной платформы, способствующей развитию программистского стиля мышления обучающихся за счет использования натурных и телесных алгоритмических средств, а также полиязыкового метода.

Становление новой ментальной учебной платформы сталкивается с рядом субъективных и объективных факторов. В первую очередь, это неготовность действующих учителей к новому формату их занятий, «неприятие» некоторыми преподавателями вуза принципов студент-центрированного обучения. Объективная причина сдерживания развития непринужденного обучения заключается в отсутствии нормативных документов вуза, регламентирующего подобный учебный процесс.

Список использованных источников

1. *Алюшин А. Л., Князева Е. Н.* Телесный подход в когнитивной науке // Философские науки. 2009. № 2.
2. Атлас новых профессий. Агентство стратегических инициатив. М.: Сколково, 2014.
3. *Найсер У.* Познание и реальность. М.: Прогресс, 1981.
4. *Нигматулина Э. А.* Особенности формирования программистского стиля мышления будущих учителей информатики // Информатизация образования и методика электронного обучения: Материалы I Международной научной конференции (27–30 сентября 2016 г., г. Красноярск). Красноярск: ИПК СФУ, 2016.
5. *Нигматулина Э. А., Степанова Т. А.* Условия формирования алгоритмической культуры студентов на основе информационного подхода // Вестник КГПУ им. В. П. Астафьева. 2011. № 1.
6. *Нигматулина Э. А., Степанова Т. А., Сокольская М. А.* Расширение понятия алгоритмического мышления при изучении современных технологий программирования в педагогическом вузе // Педагогический профессионализм в образовании: Материалы VIII Международной научно-практической конференции (16–17 февраля 2012 г., г. Новосибирск) под науч. ред. Е. В. Андриенко. Ч. 2. Новосибирск: Изд-во НГПУ, 2012.
7. *Пак Н. И.* Гипермозг как основа становления ментальной дидактики // Интернет — свободный, безопасный, образовательный: Межрегиональная научно-практическая конференция (18–19 октября 2013 г., г. Омск). Омск: Полиграфический центр КАН, 2013.
8. *Пак Н. И.* От классно-урочной системы к кластерному образованию: образовательная технологическая платформа «Мега-класс» // Международная научно-практическая конференция «Информатизация образования — 2016» (14–17 июня 2016 г., г. Сочи). М.: Изд-во СГУ, 2016.
9. *Пак Н. И.* Пространственно-временная информационная модель памяти // Фундаментальные науки и образование: Материалы I Международной научно-практической конференции (г. Бийск, 29 января — 1 февраля 2012 г.). Бийск: АГАО им. В. М. Шукшина, 2012.
10. *Пак Н. И.* Студент-центрированное обучение в образовательных кластерах // Информатизация образования и методика электронного обучения: Материалы I Международной научной конференции (27–30 сентября 2016 г., г. Красноярск). Красноярск: ИПК СФУ, 2016.
11. *Первин Ю. А.* От операционного стиля мышления через педагогические компетенции к универсальным учебным действиям. http://ito.edu.ru/sp/SP/SP-0-2010_11_23.html
12. *Hannafin M. J., Hannafin K. M.* Cognition and student-centered, web-based learning: Issues and implications for research and theory // Learning and instruction in the digital age. Springer US, 2010.
13. *Lee S., Park Y.* Customization of technology roadmaps according to roadmapping purposes: overall process and detailed modules // Technology Forecasting & Social Change. Vol. 75. 2005.
14. *Wright G. B.* Student-Centered Learning in Higher Education // International Journal of Teaching and Learning in Higher Education Vol. 23. No. 3. 2011.