

К. А. Баженова, О. В. Знаменская,

Сибирский федеральный университет, г. Красноярск,

С. В. Ермаков,

Центр современного профессионального образования, г. Красноярск

УРОВНЕВАЯ МОДЕЛЬ ОСВОЕНИЯ ПРЕДМЕТНОГО ДЕЙСТВИЯ НА МАТЕРИАЛЕ ПРОГРАММИРОВАНИЯ

Аннотация

В статье предлагается деятельностный подход к построению содержания обучения программированию на основе уровневой модели освоения предметного действия. Подход позволяет организовать обучение так, что дает ученику возможность освоить уровни организации мышления и деятельности программиста. Модель конкретизирована как серия уровневых заданий для шестиклассников по освоению понятий «алгоритм» и «программа».

Ключевые слова: индивидуальный прогресс учащихся, уровневая модель освоения предметного действия, программирование как вид деятельности.

Существующая ситуация в профессиональном программировании характеризуется двумя формально взаимно противоположными тенденциями:

- катастрофическое (гиперэкспоненциальное, по оценке экспертов издания «Computerra Online»: <https://elmcip.net/publisher/computerra-online>) усложнение информационных систем, связей и отношений, обусловленное сложностью организации распределенных многопользовательских сред;
- редукция большинства классических задач, используемых в том числе в массовом обучении программированию, к использованию стандартных инструментов разработки информационных сред и прикладных библиотек.

Уже это означает, что современное обучение программированию требует:

- организации работы в команде с удержанием системной иерархии разрабатываемой или сопровождаемой информационной системы;

- быстрого освоения инструментов разработки и библиотек с пониманием высокой вероятности их обновления и старения.

Остается актуальной проблема «человеко-машинного интерфейса», которая была сформулирована еще в 1960-х годах экспертом и разработчиком информационных систем Дж. Вейценбаумом [3] и писателем-фантастом С. Лемом [8]. Традиционно она решается попытками приспособления интерфейса к повседневным моделям человеческого поведения и восприятия, но это приводит к своеобразному антропоморфному восприятию неантропоморфных систем — к стрессам и даже катастрофам, когда эти системы начинают вести себя не антропоморфно, в своей, заданной программой, логике. Следовательно, более продуктивно было бы в рамках массового образования осваивать схемы мышления, позволяющие предсказать поведение информационных систем и использовать их максимально эффективно. Таким образом, возникает острая необходимость массового формирования информационной грамотности,

Контактная информация

Баженова Ксения Анатольевна, канд. пед. наук, доцент, доцент кафедры информационных технологий обучения и непрерывного образования Института педагогики, психологии и социологии Сибирского федерального университета, г. Красноярск; *адрес:* 660041, г. Красноярск, пр-т Свободный, д. 79; *телефон:* (391) 246-99-31; *e-mail:* mailkseniya@gmail.com

Знаменская Оксана Витальевна, канд. физ.-мат. наук, доцент, доцент кафедры информационных технологий обучения и непрерывного образования Института педагогики, психологии и социологии Сибирского федерального университета, г. Красноярск; *адрес:* 660041, г. Красноярск, пр-т Свободный, д. 79; *телефон:* (391) 246-99-31; *e-mail:* znamoksana@gmail.com

K. A. Bazhenova, O. V. Znamenskaya,
Siberian Federal University, Krasnoyarsk,

S. V. Ermakov,

Centre of Professional Education Development, Krasnoyarsk

THE LEVEL MODEL OF MASTERING OF CULTURAL ACTION PATTERN ON THE EXAMPLE OF PROGRAMMING

Abstract

The article proposes the activity approach to content-based training of programming based on the level model of mastering of cultural action pattern. This approach enables learner acquire levels of thinking organization and programming activities. The model is implemented as a series of level assignments for six-grade pupils aimed at acquiring the conceptions of algorithm and program.

Keywords: individual progress of student, level model of mastering of cultural action pattern, programming as kind of activity.

своего рода «интуиции программиста», связанной с пониманием внутренней логики работы программируемых устройств, используемых как в решении профессиональных задач, так и в быту.

В связи с этим принципиальная значимость использования в образовании заданий, требующих самостоятельного продуктивного действия, определяется тем, что ученик непосредственно видит и может оценить результат собственного действия — вместо внешней оценки учителя в традиционных задачах. **Продуктивное действие мы понимаем как «способность и готовность анализировать проблемные ситуации, моделировать схемы деятельности, строить заведомо не существующие или неизвестные схемы деятельности и действовать по этим схемам»** [5]. Согласно уровневой модели становления предметного действия, разработанной в рамках культурно-исторической концепции Л. С. Выготского, продуктивное предметное действие проходит в своем становлении три стадии, переход между которыми осуществляется путем освоения более высокого уровня средств, организующих деятельность [7].

В данной статье мы рассмотрим конкретизацию уровневой модели освоения предметного действия, подробно описанной в [7], на материале программирования. Программирование как предметный материал ценно тем, что сам характер деятельности предполагает создание конечного продукта, существующего независимо от автора: программа решает поставленную задачу при помощи формального исполнителя (вычислительной системы, робота или хотя бы даже человека, но действующего строго по правилам).

Представленный в статье подход к построению содержания обучения программированию дает ученику возможность освоить на собственном опыте специфику программирования как вида профессиональной деятельности. Акцент делается не на отдельных технологиях, моделях и платформах программирования и тем более не на освоении конкретных приемов обращения со средами разработки или прикладным программным обеспечением, а на формировании ключевых понятий и принципов, на которых основана деятельность программиста.

В ряде работ, посвященных образованию в сфере computer science, в том числе в работах Л. Л. Босовой, Н. Д. Угриновича, И. Г. Семакина и др., лежащих в основе массовых подходов к обучению основам программирования и информатики в основной школе, программирование понимается как искусство управления машинами обработки информации.

В то же время существует более широкий подход. Он представлен как в работах, посвященных специфике профессионального мышления программиста (в отличие от мышления математика или инженера) и профессиональному обучению программированию (например, таких авторов, как Э. Дейкстра, Н. Вирт [4], А. Д. Перлис, Н. Н. Непейвода [12]), так и в работах, посвященных формированию культуры программирования как таковой (С. Пейперт, А. П. Ершов и др.), как значимой части современной культуры в целом. Данный подход связан с пониманием того, что сама по себе обработка информации и выбор ее представления инструментальны по отношению к решению класса практических задач (задач

научных, инженерных, экономических вычислений, обработки текстов и изображений, управления техническими устройствами и т. д.).

Следовательно, первым шагом является превращение класса практических задач (часто решаемых на уровне интуиции, несистематизированных практических приемов) в формальные модели и задачи преобразования этих формальных моделей, представленных структурами данных и правилами преобразования.

Следующий шаг — описание последовательности преобразований на основе известных правил и свойств структур данных, формирование алгоритма решения задачи.

И уже на третьем шаге требуется превращение алгоритма в описание последовательности действий, будь то инструкция для пользователя информационной системы или набор операций, выполняемых автоматическим устройством.

Отсюда вытекает **представление о трех уровнях мышления в программировании:**

- 1) создание прикладной формальной модели и ее описание в языке структур данных, постановка задачи в терминах формальной модели (моделирование);
- 2) описание последовательности преобразований формальной модели, в совокупности решающих задачу (алгоритмизация);
- 3) написание непосредственной последовательности инструкций для исполнителя (кодирование; именно этот уровень в массовом сознании традиционно считается программированием).

Отметим, что традиционно предмет информатики появляется перед учениками как решение упражнений по алгоритмизации традиционных задач арифметических вычислений; освоение конкретных приемов обращения со средами разработки или прикладным программным обеспечением. Э. А. Нигмагулина считает, что «традиционно обучение программированию осуществляется на учебных примерах и задачах, линейно — от простого к сложному» [13]. Уровень сложности задачи задается либо количеством операций, которые подлежат выполнению в стандартной ситуации, либо сложной структурой условия, делающей неочевидной применение того или иного алгоритма решения. Лишь задачи олимпиадного типа требуют исследования объекта и применения правил, выходящих за рамки школьной программы.

Подобные методики обучения предполагают воспроизводство и передачу известных приемов действия с заранее заданным материалом, требование к усвоению значительных объемов фактической информации, например, об интерфейсах прикладных инструментов и сред разработки. А именно эта информация устаревает чрезвычайно быстро, в отличие от теоретического знания, общих схем организации деятельности, и с большой вероятностью оказывается бесполезной к окончанию процесса обучения. Решение задач в этом случае состоит в применении запомнившихся правил, определений, в ходе которого у ученика формируются такие качества, как исполнительность, внимательность, способность к волевым усилиям, — то, что определялось как основной личностный результат образования классиками, от Дж. Локка до И.-Ф. Гербарта.

В то же время в современном мире, по оценкам международных и отечественных экспертных центров (таких, как Future work skills 2020 [15], международные компетентностные состязания WorldSkills, включающие «детские» JuniorSkills и FutureSkills, Агентство стратегических инициатив РФ), на передний план выходят другие универсальные качества. Это, в частности, способность к осмыслению, выделению главного, постановке задач, социальный интеллект, инновационное и аналитическое мышление, междисциплинарная грамотность [14].

В основе освоения таких компетенций лежат два ключевых параметра:

- переход от освоения фактической информации к владению знанием технологии;
- переход от умения выполнять стандартные задания в рамках установленных правил к способности действовать в нестандартной ситуации, синтезируя новое знание.

Соотнося это с описанием структуры деятельности программиста, можно утверждать, что компетентностный подход, связанный с включением человека в продуктивную деятельность или хотя бы в ее имитацию в форме выполнения учебных заданий, весьма адекватен именно для таких дисциплин, как computer science (как и для дисциплин, связанных с прикладными инженерными разработками).

Для становления интеллектуальных умений обучающихся необходим деятельностный подход к обучению, направленный как на формирование ключевых понятий программирования, развитие мышления в процессе освоения новых знаний, так и на становление продуктивных действий по преобразованию изучаемого предметного материала. Результатом такого обучения становится целостное видение предмета не как набора знаний о нем, а как системы принципов и средств, регулирующих в том числе построение принципиально нового содержания (научных открытий, инженерных разработок). В рамках такого подхода ученики получают возможность освоить на собственном опыте специфику программирования как вида деятельности.

В качестве базовой модели для организации освоения деятельности программирования была выбрана **уровневая модель проектирования обучения, обеспечивающего индивидуальный прогресс учащихся при освоении предметного содержания**. При этом индивидуальный прогресс понимается как выход на более высокий уровень освоения предметных действий, задающийся типом их опосредования (Б. Д. Эльконин, Б. И. Хасан, А. М. Аронов, О. В. Знаменская и др.) [6, 7, 11].

Согласно указанной модели любое предметное действие в своем становлении проходит три уровня освоения.

Первый уровень — освоение общего смысла и формы действия. На этом уровне действие опосредуется правилом, шаблоном, известным алгоритмом и правильно выполняется только в стандартной ситуации.

На втором уровне ориентировочной основой действия является существенное отношение, общий принцип выполнения действия. Ориентировка на общие принципы и способы действия позволяет осуществлять его в так называемой зашумленной

ситуации, когда применение правил и алгоритмов неочевидно.

Переход на **третий, самый высокий уровень** освоения действия характеризуется включением обобщенного способа действия в состав личных ресурсов учащегося. Действие по сути становится продуктивным, таким, что обобщенный способ, заимствованный в известном предмете, уместен и выступает как преобразующий материал для другого предмета. Это позволяет в новых ситуациях, отличных от ситуации формирования, принимать и отвергать, корректировать и преобразовывать само существенное основание способа действия, требует умственных действий рефлексии, анализа, синтеза и обобщения высокого уровня [7].

На основе уровневой модели разработан инструментарий «Дельта» для диагностики уровня индивидуального прогресса при изучении русского языка и математики в начальной и основной школах (О. В. Знаменская, О. И. Свиридова, Л. А. Рябина и др.) [6, 7, 9]; тест SAM [11].

Реализуются программы повышения квалификации для педагогов по освоению указанных инструментариев и проектированию условий обучения, способствующих индивидуальному прогрессу школьников и достижению личностных, предметных и метапредметных результатов (К. А. Баженова, О. И. Дятлова, О. В. Знаменская, О. И. Свиридова, Л. А. Рябина и др.) [6, 7].

Образцы учебных и диагностических материалов, основанных на уровневой модели, разработаны для школьных учебных предметов «Математика», «Русский язык», «Физика», «Биология». Пионерские работы, посвященные созданию учебных и диагностических материалов по математике, основанных на уровневой модели, создавались коллективом авторов по заказу НФПК (А. М. Аронов, О. В. Знаменская, К. А. Баженова, В. Г. Ликонцева, О. А. Францен) [7, 9]. Заметим, что в школьной математике также выделяется алгоритмическая линия, однако она не охватывает систему ключевых понятий и предметных действий информатики и программирования.

Мы предполагаем, что **применение уровневой модели как средства анализа структуры деятельности программирования позволяет разработать компетентностную модель обучения программированию в рамках изучения курса информатики**.

Конкретизируем уровни освоения предметного действия, учитывая специфику программирования как вида профессиональной деятельности.

Первый уровень. Воспроизведение последовательности действий по образцу или отработанной схеме, в идеале доведенное до автоматизма. Учащийся может выполнять конкретные операции (возможно, достаточно сложные) для решения стандартно поставленной задачи по готовому алгоритму с использованием стандартных процедур. Он частично может опознать условия, в которых выполнение последовательности действий затруднено.

При освоении деятельности программирования задачи первого уровня связаны с освоением базовых процедур кодирования и алгоритмизации, правил использования хотя бы одного языка программирования: прежде чем браться за сложные задачи, ученик должен освоить хотя бы один инструмент для решения этих задач. Учебные задания этого уровня могут

быть направлены на составление алгоритма для другого исполнителя-человека и понимание различий исполнителя-машины и исполнителя-человека.

Второй уровень. Решение класса задач, ориентированных на построение действия на основе комбинации образцов для решения поставленной задачи, включая анализ условий и отбор необходимых средств. Обучающийся может формулировать задачу в результате анализа проблемной ситуации, выбирать способ, соответствующий условиям и ограничениям задачи, а также оценивать адекватность результата действия.

Учебные задания этого уровня требуют умения конструировать и комбинировать алгоритмы, переводить их в код, обосновывать работоспособность алгоритма и его эффективность для решения поставленной задачи. Действие на этом уровне предполагает выделение общего принципа работы с понятием «программа» и обнаружение существенных различий между понятиями «алгоритм» и «программа». На этом этапе ученик владеет семантикой языков программирования, может разбивать задачу на подзадачи, строить сложную программу как совокупность более простых, использовать библиотеку алгоритмов.

Н. Н. Непейвода отмечает, что «практическая цель обучающегося — научиться даже без предварительного изучения языка видеть в нем общие конструкции» и «понимать программы, написанные на нем» [12]. Таким образом, в качестве *предметного результата обучения* предполагается достижение второго уровня освоения предметных действий.

Третий уровень освоения деятельности программирования предполагает конструирование способа для решения задачи в условиях, когда способ отсутствует или неизвестен. Обучающийся способен комбинировать и трансформировать освоенные способы для решения проблемы, используя теоретические знания и объективированный опыт, превращать проблему в задачу, формулировать подзадачи на основе анализа проблемной ситуации, выстраивать последовательность их решения, определять уровень сложности задачи, сходство и различие с известными типами задач.

Учебные задания этого уровня могут требовать использования известных алгоритмов и языков программирования для решения прикладных задач моделирования в конкретной предметной области. Особенностью задачного материала для освоения третьего уровня предметного действия является необходимость оперировать сведениями из той предметной области, в которой изначально поставлена задача. Таким образом, *продуктивное действие требует удержания межпредметных связей, необходимых для решения задачи.*

В качестве примера системы заданий, разработанных на основе уровневой модели, приведем серию заданий, направленных на освоение понятий «алгоритм» и «программа», созданных при участии К. В. Морозовой [10]. Система заданий устроена подобно заданиям алгоритмической линии диагностики «Дельта» по математике [9]. Задания могут быть использованы учителем как для диагностики уровня освоения шестиклассниками предметного действия, так и для формирования указанных понятий.

Серия заданий содержит предваряющий текст, два задания первого уровня, два задания второго

уровня и три задания третьего уровня, а также две двухуровневые задачи, уровень которых определяется в зависимости от способа решения, предъявленного обучающимся.

В предваряющем тексте описывается сюжет серии заданий, приводятся необходимые для выполнения заданий алгоритмы. Выполнение заданий требует от обучающихся удержания базовых характеристик алгоритма (результативность, дискретность, детерминированность) и выделения существенного условия перехода от алгоритма к программе — наличие формального исполнителя. Деятельность программирования изначально связана с решением прикладных задач, постепенным отчуждением алгоритма действий от исполнителя-человека и пониманием существенных условий реализации программы, постепенной передачей инструкций от неформального исполнителя (человека) к формальному (машине).

Задания первого уровня (1.1 и 1.2 в [10]) выявляют, понимает ли учащийся ограничения формального исполнителя (компьютера) по сравнению с неформальным исполнителем (человеком). Первому уровню соответствует такое решение обучающихся, когда учащийся может воспроизводить алгоритм в соответствии с образцом. Посредником при решении задач являются образец, правило, шаблон или алгоритм. При выполнении задания не требуется выходить за рамки образца, достаточно формального следования указанным действиям. Задание считается выполненным, если правильно выполняется обучающимися исходный алгоритм.

Задание 1.1. Нарисуй букву по алгоритму, написанному для чертежника (предлагается первая четверть координатной плоскости на клетчатой бумаге): Сместись в точку (3;1); Опустить перо; Сместись в точку (3;4); Сместись в точку (5;4); Сместись в точку (5;1).

Задание 1.2. Напиши программу «Буква Б», которая изображена на координатной плоскости (предлагается место для ответов и рисунок с изображением алгоритма перемещения пера чертежника).

Двухуровневые задания (задания 1.2.1, 1.2.2 в [10]) могут быть выполнены как на первом, так и на втором уровне [7]. Многоуровневое задание считается выполненным на первом уровне, если учащийся использует алгоритм как неформальный исполнитель, не учитывая того, что алгоритм необходимо составить для исполнителя-чертежника. Задание выполнено на втором уровне, если учащийся при обосновании своего выбора опирается на анализ выполнения действий формальным исполнителем.

Задание 1.2.1. Нарисуй элементы буквы Ю. Какие еще буквы можно нарисовать из этих элементов? Напиши алгоритм для чертежника, в состав которого входит изображение одного из элементов буквы Ю. Придумай несколько вариантов. (Предлагается клетчатое поле.)

Задание 1.2.2. Предлагается алгоритм для исполнителя-человека. Напиши алгоритм для чертежника (алгоритм состоит из 10 элементарных действий, которые может выполнять чертежник [10]). Отличаются ли алгоритмы для человека и для исполнителя? Чем?

Решение заданий второго уровня (см. задания 2.1 в [10]) предполагает понимание учащимся различий между алгоритмом и программой. Задания второго уровня позволяют определить, способен ли ученик обнаружить последовательность шагов алгоритма; оперировать полученной информацией о наборе команд, которые может выполнять исполнитель; представить образ будущего результата после завершения алгоритма; выделить общий способ действия написания букв исполнителем чертежником.

Задание считается выполненным на втором уровне, если правильно и до конца выполнен алгоритм по заданной инструкции в новых условиях, выходящих за рамки применимости инструкции, и обнаружено существенное отличие условий, не позволяющее непосредственно применить освоенную инструкцию ранее.

Задание 2.1. *Узнай, какую букву можно начертить, действуя по алгоритму (дан алгоритм [10], представляющий последовательность элементарных действий, два из которых пропущены. Обучающийся должен реконструировать их сам). Восстанови недостающие действия. Сможет ли чертежник решить эту задачу? Получит ли он тот же результат, что и ты? Обоснуй свой выбор.*

Решение заданий третьего уровня (например, заданий 3.1, 3.2 в [10]) предполагает использование кодирования для разработки программы, отвечающей определенным условиям (например, алгоритм должен быть оптимальным по количеству операций). Выполнение задания требует применения общего принципа построения буквы по ее составляющим к созданию части кода в алгоритме для отдельно взятых элементов и удержания алфавита как системы элементов. Ученик, выполняющий задания третьего уровня, способен не только видеть границы применения нового алгоритма, заданного ранее образцом, но и трансформировать этот алгоритм в соответствии с изменениями в материале.

Задание 3.1. *Нужно научить чертежника рисовать буквы Р, А, Ф. Напиши общую часть алгоритма для чертежника так, чтобы ее можно было использовать для всех этих букв. Добавь к готовому алгоритму дополнительные шаги для изображения каждой буквы по отдельности.*

Задание 3.2. *Напиши программу, которая рисовала бы буквы с помощью фрезеровочного аппарата. Составь базу элементарных элементов для написания букв.*

Предлагаемый авторами подход к анализу структуры деятельности программирования позволяет выделить ключевые понятия («алгоритм» и «программа»), которые необходимы при изучении информатики. Использование уровневой модели [7, 10] дает основу для проектирования серии уровневых заданий на освоение указанных понятий.

Рассмотренная система задач может быть использована как при конструировании содержания обучения в деятельностном подходе, так и в качестве материала для выявления типа и интенсивности индивидуального прогресса обучающихся при освоении информатики.

Деятельностный подход к формированию понятия «программирование» на основе уровневой модели создает базу для построения метапредметных понятий и, следовательно, для становления ключевых способностей человека к анализу нетипичной ситуации и конструированию способа действия в ней. Смещение акцента на формирование ключевых понятий и принципов информатики (computer science) при постановке проблем подготовки учителей информатики [2] позволяет расширить рамки методики преподавания информатики и требует деятельностного подхода к осмыслению проблемы.

Список использованных источников

1. Абельсон Х., Сассман Дж. Дж. Структура и интерпретация компьютерных программ. М.: Добросвет, 2004.
2. Аронов А. М., Баженова К. А. Современная технология формирования деятельностного отношения к образовательному процессу у студентов — бакалавров педагогики // Известия Южного федерального университета. Педагогические науки. 2015. № 11.
3. Вейценбаум Дж. Возможности вычислительных машин и человеческий разум. От суждений к вычислениям. М.: Радио и связь, 1982.
4. Вирт Н. Алгоритм + структуры данных = программа. М.: Мир, 1985.
5. Ермаков С. В., Попов А. А. Дополнительное математическое образование как условие развития математической одаренности. <http://opencu.ru/page/dopolnitelnoe-matematicheskoe-obrazovanie-kak-uslovie-razvitiya-matematicheskoy-odarennosti>
6. Знаменская О. В., Баженова К. А., Скрипка А. М. К построению схемы анализа образовательных условий индивидуального прогресса учащихся // Теоретические и прикладные проблемы науки и образования в 21 веке: Сборник научных трудов по материалам Международной заочной научно-практической конференции: в 10 частях. Тамбов: Консалтинговая компания Юком, 2012.
7. Знаменская О. В., Свиридова О. И., Рябинина Л. А. Оценка-поддержка индивидуального прогресса учеников. Красноярск: СФУ, 2014.
8. Лем С. Сумма технологий. М. — СПб.: АСТ — Terra Fantastica, 2002.
9. Мониторинг индивидуального прогресса учебных действий школьников. Красноярск: Печатный центр КПД, 2006.
10. Морозова К. В. Серия уровневых заданий по программированию для обучающихся 6-х классов на основе модели индивидуального прогресса. Красноярск: СФУ, 2016. <http://elib.sfu-kras.ru/handle/2311/28701>
11. Нежнов П. Г., Карданова Е. Ю., Рябинина Л. А. Исследование процесса присвоения учебного содержания // Вопросы образования. 2013. № 4.
12. Непейвода Н. Н. Основания программирования. http://people.toiit.sgu.ru/Sinelnikov/PT/Nepeivoda/nepeivoda_prog.pdf
13. Нигматулина Э. А. Условия формирования алгоритмической культуры студентов на основе информационного подхода // Вестник Красноярского государственного педагогического университета им. В. П. Астафьева. 2011. № 1.
14. Попов А. А., Ермаков С. В., Реморенко И. М. Проект «Оценка компетентностных результатов и достижений» // Открытое образование как практика самоопределения. М.: Некоммерческое партнерство «Авторский клуб», 2015. <http://opencu.ru/uploads/proekt.pdf>
15. Future work skills 2020. Report SR-1382A // Institute for the Future; the University of Phoenix Research Institute. <http://www.iftf.org/our-work/global-landscape/work/future-work-skills-2020>