

## **Состояние и перспективы изучения информатики в школе: вопросы концептуализации, онтологии и таксономии знаний, получаемых при обучении.**

Николае ПЕЛИН - Тираспольский государственный университет (Кишинэу)

### **Summary**

This article is devoted to the analysis of the problem related to the content and quality of informatics study in school (gymnasium, lyceum). The list of manuals recommended by the Ministry of Education for grades 7-12 are analyzed. The study program content, that the lecturer should guide is a subjected of critical analysis. The author expresses his opinion regarding the level of study material assimilation by former pupils on Informatics, based on the analysis of the conversations and knowledge verification of some students groups. It was expressed the vision regarding Informatics training concept for school pupils, according to contemporary conditions.

**Keywords:** education, school, computer science, information, programming methods, computer

### **Rezumat**

Articolul este consacrat analizei problemei ce ține de conținutul și calitatea însușirii informaticii în școală (gimnaziu, liceu). Sunt analizate manualele recomandate de către ministerul educației pentru clasele 7-12. A fost expus analizei critice conținutul programului de studii după care trebuie să se conducă lectorul. Autorul și-a expus părerea privind nivelul de însușire a materialului la disciplina „Informatică” predat în școală după analiza rezultatelor a verificării cunoștințelor studenților sau conversațiilor cu ei. De asemenea, a fost expusă viziunea sa privind conceptul de instruire a disciplinei „Informatică” pentru elevii școlilor în condițiile contemporane.

**Cuvinte-cheie:** instruire, școală, informatică, informație, metode de programare, calculator.

### **Введение.**

Проблема состояния дел с изучением и организацией изучения отдельных разделов курса Информатики в школах Республики Молдова автором настоящей статьи достаточно внимательно исследовалась в конце 80-х, начале 90-х годов прошлого столетия, с целью предложить, на инициативных началах, вариант концепции школьного курса по информатике. О судьбе этого варианта концепции уже писалось в [1,2]. Впоследствии пришлось переключиться на другие вопросы. Возможно, никогда более к проблеме школьного курса по Информатике возврата не было бы, если бы постоянно не смущал недопустимо низкий уровень знаний информатики молодых людей, приходящих на обучение в университеты, несмотря на улучшающуюся от года к году техническую базу, некоторые изменения к лучшему в школе и семье. Но, окончательно побудили к написанию настоящей статьи три случая. На взгляд автора, они как-то связаны. Они также наводят на мысль что, что-то или кто-то школьным учителям, учёным и специалистам мешает излагать в научно - методических изданиях свои соображения по поводу программы обучения информатики в школе, её улучшения. Мешает проявлять инициативу и выдвигать новые (иные, и по содержанию, и по методологии изучения) концепции обучения, предлагать иные программы, учебники и т.д., чтобы добиться лучших результатов в обучении. Во всяком случае, дискуссия на тему качества учебно-методической базы и программы обучения информатики в школе не помешает. По многолетним наблюдениям, проводимых автором этой статьи, в лучшем случае, 15-20 % выпускников лицеев, которые более - менее осваивают пройденный материал по информатике. А это очень мало. Да и у тех далеко не цельное (системное) представление об информации, информационных процессах, компьютере, а какие-то обрывки знаний.

Статья написана на русском языке, в надежде охватить большее количество читателей. На румынском языке эту статью и статью, изданную ранее в [2], также можно будет найти несколько позже на сайте [www.prolog.md](http://www.prolog.md) [3].

### **Но сначала о случаях побудившие к написанию статьи.**

**Случай первый:** Несколько лет назад, в одном из университетов Республики Молдова, был почему-то снят из учебного плана курс логического программирования. Точнее, курс был сдвинут из числа обязательных курсов в число необязательных (optional), да ещё и с уменьшением количества учебных часов. Случай вроде бы обычный, учебные планы достаточно часто изменяются. Но, чтобы снять логическое программирование, изучая которое студент, обучающийся по профилю Информатика, должен получить минимально необходимое представление о декларативном программировании, - это особый случай. Дело в том, что все языки программирования делятся на две большие группы: декларативные и процедурные языки программирования, кардинально отличающиеся и дополняющие друг друга. Это как у плотника есть левая и правая рука, как без одной из рук выполнить свою работу хорошо. О них (декларативных и процедурных языках программирования) подробнее будет изложено ниже. Кроме того, человек, программирующий в логике, постоянно развивает своё логическое мышление. Да и курс по логике, из школьной программы, почему-то убрали. Утверждать, что у студентов наших ВУЗов нет проблем с логикой мышления, было бы не совсем правильно. Точнее было бы утверждать, что развивать логику мышления необходимо всегда, в любом возрасте, всем, при любых возможностях.

**Случай второй:** К 45-летию появления первого интерпретатора языка логического программирования Пролог и 25-летию событий, которые повлекли к не появлению в курсах по Информатике в школе, разделов по изучению декларативного, в частности логического языка программирования, автор представил к опубликованию статью, в одном из научно-методических журналов, издаваемых в Республике Молдова. Посвящена была роли логического программирования в изучении информатики. Но эта статья была отклонена «неизвестным» автору оппонентом с обоснованием, примерно следующего содержания: «статья посвящена программированию, но автор ничего нового в своей статье не предлагает, всё, о чём он пишет, широко освещалось в разных публикациях 80-х годов, поэтому эту статью публиковать не следует». Впоследствии эта же статья без изменений была опубликована в [2]. Имея доступ к ней, читатель может заметить, что, содержание статьи и акценты на что концентрируется внимание рецензента в рецензии, несколько расходятся. Зачем? Может рецензент всё же прав в своей оценке, может, направленность статьи не настолько очевидна? А может и что-то другое.

**Случай третий:** На одном из научных семинаров, претендент на соискание ученой степени представлял результаты интересных исследований относительно применения новейших технических средств обучения (ТСО) для преподавания курса «Tehnici de programare» (Методы программирования). Результаты исследования в применении конкретных ТСО сомнений не вызывало, но озадачило содержание курса, для изложения которого предполагалось применить эти результаты. Практически содержание его в главном совпадало с содержанием учебника [4]. Но и в этом возможно не может быть вопросов к претенденту. Возможно, молодой учёный тоже придерживается такой же точки зрения, как и автор учебника [4] и они правы, а не прав автор данной статьи. Возможно, претендент осознаёт частность [4], выдаваемого за общее, но обстоятельства вынуждают воздержаться пока от своих комментариев. Кроме того, есть подходящий вариант выхода из ситуации. В качестве объекта приложения разработанной им теории может быть взят объект, признанный в качестве адекватного авторитетным органом в Республике. Так что к претенденту, во всех аспектах, кажется, не может быть претензий,

но на системном уровне, в организации исследований в республике, некий пробел всё-таки просматривается. Почему? Проанализируем на этом же примере. Взяв в качестве объекта для своих исследований, программу курса «Tehnici de programare» (Методы программирования) в близком соответствии с содержанием школьного учебника [4], и не подвергнув его предварительно критическому анализу, косвенно признаётся безукоризненность этого прототипа. Но, при всём уважении к Коллегии министерства образования утвердившего учебник [4] и его автору, его известности, методы программирования и методы разработки алгоритмов, их анализ - совсем разные вещи. Аналогичные названия учебников мы находим в Румынии, например в учебнике автора Tudor Sorin, тут же в названии ставит соответствующее ограничение: "Informatica (Tehnici de programare) Varianta Pascal". Manual pentru clasa X-a. Respectă noua programă valabilă începînd cu anul 2000. Ограничением служит, именно название языка. Да и последняя фраза, явно подчёркивает возможность альтернатив. А в [4] нет, здесь охват бескрайний и этим, словосочетание "Tehnici de programare" (методы программирования), приобретает иное истинностное значение, имея в содержании описание только, анализ алгоритмов и методы их разработки, что достаточно может быть для освоения процедурного программирования, но не более. И альтернативность в [4], для молодого исследователя вряд ли представляется несомненной, при грозной пометке на 2-й странице: «Aprobat de către Colegiul Ministerului Învățămîntului ...».

Тогда как же быть будущим учёным, когда с одной стороны – необходимость точного описания истины, с другой утверждённые министерством образования учебники, в которых однозначно очерчивается содержание курса. Совет, - придерживаться философии крылатой фразы: «Платон мне друг, но истина - дороже». Да верно, но не всегда рядом с нами такие люди как Платон. А что случилось с автором этой фразы, после смерти Платона? Был сразу же уволен из академии. Не каждый рискнёт пойти по аристотелевскому пути, имея реалии сегодняшнего дня. Проанализированный выше случай подсказывает (допускает возможность), иногда авторитет автора книги и авторитет органа его утвердившего – может стать дополнительной для молодых исследователей проблемой. Ещё хуже если они (молодые ученые) могут впасть в зависимость от тех лиц, которые заинтересованы в сохранении без изменений учебной программы или придерживаются той или иной точки зрения из-за своих заблуждений. Например, если вдруг, кто-то из них станет официальным оппонентом их научных работ.

Эти и иные причины послужили основанием считать, что что-то или кто-то (вольно или невольно) могут тормозить публичный анализ и обновление учебных программ, препятствовать появлению и иных альтернативных учебников для изучения информатики в школе, научно-методических публикаций на эту тему. Поэтому настоящая статья необходима, пусть и не безупречна, актуальна и может послужить началом исследований, нацеленных на устранения возможных несоответствий рекомендованных Министерством образований учебников и программ обучения в школе курса по Информатике.

### **Состояние проблемы.**

В результате проведенного анализа учебников по информатике [4-11], рекомендованных в установленном порядке министерством образования Республики Молдова в качестве учебников для 7-12 классов под общим названием Информатика, а также соответствующих пособий для учителя, учащиеся разных поколений гимназистов и лицейстов уже много лет изучают следующее:

- ученики 7-х классов – Структуру компьютеров, операционные системы и редактирование текстов по учебнику [9] изданному (переизданному) в 2001, 2003 и 2005гг.;
- ученики 8-х классов – хранение информации на рабочих листах, формулы и вычисления, диаграммы и графические объекты, диаграммы и графические

- объекты, базы данных и алгоритмы по учебнику [5] изданному (переизданному) в 2005г.;
- ученики 9-х классов – язык Паскаль по учебнику [6] изданному (переизданному) в 2006г.;
  - ученики 10-х классов – по учебнику [7], изданному (переизданному) в 2000, 2001, 2002 гг. учебнику с общей тематической направленностью «Структура компьютера» (?!), разделы: составные типы данных языка Паскаль, фундаментальным знаниям из теории информации, арифметические основы вычислительной техники, булеву алгебру, логические схемы, устройство и работа компьютера, компьютерные сети. В 2007 г. издан учебник [8], в котором, уже отсутствует глава, посвященная составным типам данных и общая тематическая направленность;
  - ученики 11-х классов - по учебнику [4] изданному (переизданному) в 2003 и 2005гг. с общей тематической направленностью «Методы программирования» (?!). Разделы: анализ алгоритмов, методы разработки алгоритмов и алгоритмы решения некоторых математических задач. В 2008 г. издан учебник [10], в котором уже отсутствует общая тематическая направленность, возможно из-за включения 3-х новых глав: функции и процедуры, динамические структуры данных, методы разработки программных продуктов;
  - ученики 12-х классов – численные методы, базы данных (Acces), HTML по учебнику [11], изданному в 2010 г. .

Учебники рекомендованы и для реального и для гуманитарного профиля. Изданы учебники на румынском и русском языках. Альтернативные варианты учебников, написанные и изданные другими авторами, предусматривающие возможность работы по этим же или другим учебным программам и утвержденными нашим министерством образования, автор данной статьи не увидел, ни на прилавках ряда книжных магазинов, ни в библиотеках учебных заведений. Возможно, автор провел не достаточно полное исследование изданных книг по данной теме. Возможно, описанный выше пакет учебников для изучения курса информатики в гимназиях и лицеях действительно весьма подходящий. Возможно, в республике нет подходящих специалистов, способных предложить лучший вариант программ обучения, написать и издать соответствующие учебники. Но также возможно, дела в стране поставлены таким образом, что альтернативы, по крайней мере, в данной проблемной области не возможны или недопустимы.

Даже если какие-то из изданий учебников, остались вне поля зрения автора, возможно, пропущены, - общая картина вырисовывается чётко. В течение 5-6 лет обучения на уроках по информатике, руководствуясь вышеперечисленными учебниками, ученики изучают структуру компьютера, операционные системы, редакторы, электронные таблицы (EXCEL), методы разработки алгоритмов и программирование компьютера с помощью этих алгоритмов на языке Паскаль, численные методы, базы данных (Acces), HTML. Вопросы, прямо или косвенно, связанные с Паскалем занимают примерно две трети всего времени. Это вся программа курса по информатике в 9, 10 и 11-м классах. Есть некоторые сведения об информации, о логических элементах, сумматорах, генераторах импульсов и другое. Впрочем, и в соседних странах можно встретить аналогичные разделы в учебниках по информатике. И каждый в отдельности раздел, с точки зрения компьютерной науки, написан неплохо.

Всё так. Только кто-то исследовал ли, после полного изучения курса по такой программе вырисовывается ли у ученика системное представление, как нечто единое целое из всего конгломерата технических средств, программ, процессов, методов, терминов и т. д.? Почему такой низкий процент выпускников школ освоивших в полном объёме программу курса? А из тех, кто всё-таки освоил в полном объёме программу курса по информатике в

школе, способны ли они эффективно применить свои знания? Имеют ли они ясное представление о информации, информационных процессах, обрабатываемых с помощью компьютеров, методах и средствах их обработки, о системах и системном подходе, об особенностях потребителя для кого предназначена обрабатываемая информация? Почему у них лучшие успехи в освоении HTML (декларативный язык разметки текстов), EXCEL (своеобразная система функционального программирования), почему достаточно скромные успехи в Паскале (язык процедурного программирования)?

В статье будут представляться аргументы усиливающие важность и необходимость изучения декларативного и, в частности логического программирования в школе, как одно из важнейших направлений в программировании (вопросы концептуализации), альтернатива процедурному программированию. С другой стороны, необходимость учёта интереса корпораций Soft, который сегодня более нацелен на применение таких процедурных языков, как C++, Java и др. (прагматический аспект). Науки информатика и педагогика очевидно требуют изучения обеих методов программирования, как для получения наиболее полной картины мира информатики (вопросы онтологии), так и для нахождения наиболее эффективной методологии, последовательности их изучения (вопросы таксономии).

### **Об объёме и содержании знаний: критический анализ текущего состояния.**

Так что же должна содержать учебная программа школьного курса по информатике, чтобы знания и умения, полученные в ходе обучения, были максимально полезными. Представленный, в учебниках для 7-го, 8-го и 12-го классов, материал - необходим. Пожалуй, только эти три учебника вызывают меньше вопросов. Может вот только вместо Access лучше дать SQL, а материал для 7-го и 8-го класса, объединить в один для 7-го класса и даже добавить ещё что-то, например PowerPoint. Необходимо учитывать факт роста компьютерной грамотности общества, в котором воспитывается подросток.

О количестве информации, её кодирования и квантования изображений не мешает знать. Знать логические элементы, их классификацию, сумматоры, генераторы импульсов и другие полезные вещи важно. Но важно также для «чего», т.е. где конкретно может применить эти отдельные элементы полученного знания, чтобы учащийся получил необходимые умения. Но даже если, допустим, получил необходимые знания и умения по построению и использованию для чего-то некоторых RS-триггеров, он (ученик), увидит его роль (триггера) и место в такой сложной системе каким является компьютер? И даже если этот триггер преподаётся только лицеистам реального профиля, не все они пойдут учиться по специальности Компьютеры. 80-90% из них, а может и более, станут инженерами строителями, механиками, и.т.д. Но тогда зачем преподавать материал, который для данной аудитории слушателей, на этом этапе обучения пока не нужен, подавляющей части в будущем не понадобится, а тем, кому понадобится в будущем, могут это изучить без всякого ущерба, будучи уже на студенческой скамье. В таком случае, какова степень полезности и важности этого материала в рамках учебной дисциплины «Информатика» на этом уровне образования? Особенно если речь идёт о гуманитарном профиле. Но, давайте не будем говорить о полезном. Сконцентрируемся на необходимом. Все ли необходимое вошло в учебную программу для получения фундаментальных представлений об обработке информации с помощью компьютера?

Автором, в течение ряда лет, перед началом чтения своих учебных дисциплин, проводится в некоторых академических группах беседа, входной контроль знаний студентов, обучающиеся на специальностях по профилю информатика. Выявляются, весьма тревожные результаты. Материал школьной программы пройденного курса по информатике, даже если знает, найти, поставить и решить задачу по реализации простой информационной системы не может. Обрывки знаний есть, но цельной картины об информации, информационных процессах, информационных системах, абстрактное представление о процессе переработки информации с помощью компьютера (а это не то

же самое - из каких конструктивных элементов построен компьютер, и принцип работы этих элементов), о компьютерной программе, методах программирования у большинства студентов нет. Написать модель для решения простой задачи, составить блок-схему алгоритма, построить функциональную схему информационной системы – проблема. Получить ответ, какие бывают парадигмы программирования и их описание – проблема. Логика мышления оставляет желать лучшего. И как не вспомнить о философии Массачусетского Технологического Института уже ряд лет входящего в пятерку лучших учебных заведений мира: «обучение (должно быть) не в манипуляции и мгновенных деталях науки, которые можно применить лишь на практике, а в знании и понимании всех основных научных принципов с их объяснениями».

### **Информатика как наука.**

Обратимся к значению слова информатика, ставшего названием научного направления. Для нас особенно не безразличны интерпретации слова, в румынской и русской языковой среде. В соответствии с [12], информатика это наука, занимающаяся обработкой информации с помощью компьютеров (систем автоматизированных вычислений). В соответствии с [13], информатика – наука о методах и процессах сбора, хранения, обработки, передачи, анализа и оценки информации с применением компьютерных технологий, обеспечивающих возможность её использования для принятия решений. Более раннее определение, приведенное в [14], дает определение информатики как дисциплины, изучающей структуру и общие свойства научной информации, а также закономерности её создания, преобразования, передачи и использования в различных сферах человеческой деятельности. Французская академия в 1967 г. дала определение информатики как науки рациональной обработки информации, особенно с помощью средств автоматизации [15].

Все определения практически одинаковы и сводятся к тому, что главенствующим понятием здесь является информация и методы её обработки, а компьютеру отводится роль инструмента, очень необходимого, но всё же инструмента. Дай инструмент в руки того, который не сумеет «увидеть» полезную информацию для конкретного пользователя, переработать её и представить по форме, содержанию, месту и времени наилучшим образом именно для этого пользователя и, этот компьютер станет не более чем дорогой игрушкой.

### **Подходы к разработке альтернативной учебной программы курса по Информатике.**

Как видно из определения информатики ученик, за годы обучения должен получить общее знание о таких понятиях как информация (**что**), методы её обработки (**как**) с помощью компьютера (**с чем**). Для получения полноценных умений в применении своих знаний ученик должен ещё изучать потенциального потребителя этой информации (**для кого**). А всё вместе уже говорит о наличии системы. Кое-какие элементы знаний из теории систем также необходимы. Иметь представление о целостности систем, об открытых и закрытых системах, о простых и сложных системах. Следует иметь представление о семантике информации в системах, тезаурусе приёмника информации (потребителя), от состояния которого зависит возможность воспринять и интерпретировать тот или иной фрагмент информации, уровнях понимания информации приёмником (потребителем). Всё это уже начинает формировать у учащегося системный подход.

Информация в информационных системах всё равно, что металл или другой материал на машиностроительном заводе. Необходимо изучать структуру информации, имеющей смысл и полезность для пользователя, имеющей в качестве элементов данные, входящая в свою очередь в другую, особо организованную структуру информационных единиц, на основе которой уже можно принимать решения, которая может уже называться

знаниями. Знать что такое жизненный цикл информации, её актуальность для потребителя. Знать каким процессам подвергается информация.

Иметь общее представление об архитектуре компьютера, из каких основных блоков состоит - необходимо. Знать, что компьютер это последовательная машина и уметь продемонстрировать его работу в самом общем виде, на так называемой машине Тьюринга. Из грамматики Хомского необходимо (хотя бы на популярном уровне) иметь представление о правиле переписывания и, её значении при разработке любых языков для программирования компьютера.

Что касается вопросов программирования компьютера, то забегая вперёд здесь, прежде всего, необходимо говорить о методах программирования и программировании на одном из декларативных языков, например для начала языка Пролог (или ему подобном), как для гуманитарного так и реального профиля, а для реалистов необходимо дать после Пролога еще один из языков, но уже из группы процедурных, например начала C++ или Java. Язык Java, прочно занимает уже несколько лет одно из первых мест в мире по числу применений. И, когда речь идёт о методах программирования, то это совершенно не то, о чем пишется в учебнике [4].

Язык программирования, подлежащий изучению, его выбор и обоснование выбора, учащийся должен иметь возможность проанализировать и знать назначение и роль (место) выбранного метода программирования для того или иного профиля обучения. Это настолько важно, а по продолжительности изучения этой части курса информатики, настолько значительно, что целесообразно провести анализ методов программирования и обоснования выбора языков программирования ниже, в отдельном разделе данной статьи.

### **Методы программирования и обоснование выбора языков для изучения.**

Полезно рассмотреть четыре языка программирования, отличающиеся значительно друг от друга по методам программирования, ставшими родоначальниками большинства современных языков и представленных в хронологическом порядке:

1. Один из самых старых языков программирования Fortran (1957 г., Джон Бэкус и др., США) и подобные ему языки процедурного программирования (Алгол, PL/1, Pascal и др.). Эти языки предназначались для решения вычислительных задач, возникающих в математике, физике, инженерных расчетах, и т.д.. Они в основном работают с числами.

2. Вторым языком после Fortran, стал язык функционального программирования LISP (1958г., Дж. Маккарти). Предназначался для работы со списками, представляющие программы и данные. Своим появлением проложил путь для декларативного программирования.

3. В 1962 г. (авторы, Оле-Йохан Даль и Кристен Нюгорд, Норвегия) разработан язык SIMULA I, первый из языков объектно-ориентированного программирования, впоследствии целый ряд ему подобных языков (SIMULA 67, Smalltalk, C++, Java и др.), где центром всего является объект, являющийся экземпляром какого-то определенного класса, а классы образуют иерархии. Но метод программирования, в основе своей, всё же остался процедурным.

4. В 1971 г. (Ален Колмероо, Франция) разработал язык логического программирования Пролог. Автор, лингвист по специальности, натолкнулся на идею реализации языка Пролог, в поисках подходящих методов обработки текстов естественного языка на компьютере. Годится и для решения вычислительных задач и для обработки естественного языка. Пролог – язык декларативного программирования.

Если верить [16] человечество придумало более восьми тысяч языков программирования и, каждый год, их число увеличивается. Но, все они, по методу программирования, могут быть разделены на две большие группы [17]:

- **процедурные** (императивные) языки программирования, где теоретической моделью служит алгоритм, который предписывает, **как** решить задачу и представить результат. По представленной выше хронологии это языки 1 и 3.

- **декларативные** языки программирования, с помощью которых описывается, **что** представляет собой проблема и ожидаемый результат. Сюда входят языки функционального (2) и логического (4) программирования.

В первой группе языков, для того чтобы предписать, **как** решить задачу (вопросы управления процессом вычислений) естественно нужно ещё знать **что** (вычислять-логику проблемы). Естественно, начинать изучение с языка процедурного программирования значит заведомо усложнить задачу обучения для учащегося, делающего первые шаги в информатику, где всё ново, ещё ничего неизведанно. Необходимо разобраться и с тем, **что** представляет собой проблема, представить её модель (а уже только это не просто, но кое-что наработано и при изучении других дисциплин), и **как**, шаг за шагом управлять процессом вычислений с помощью компьютера (а здесь уже всё ново).

С чего начинать изучать программирование в рамках учебной программы курса информатики в школе? Вообще то, логично, от простого - к сложному, с учетом профиля обучения. Но, давайте подумаем, насколько необходим гуманитариям язык процедурного программирования? Наверное, не помешает. А вот декларативный язык для гуманитарного профиля наверняка более необходим. А достаточно ли для реалистов ознакомление только с одним единственным языком программирования? Чтобы уложиться по времени в заданное число часов для курса Информатика, всё зависит глубины обучения языка. По мнению автора, для реального профиля было бы, лучше изучать язык программирования не углубленно как сейчас изучается Паскаль, а дать только основы программирования, но на двух разных языках программирования, обязательно из разных групп. Если войдёт во вкус программирования хотя бы на одном из языков, ученика уже не возможно будет остановить, он сам изучит необходимое, выходящее за рамки основ. И главное, при преподавании языка давать не предикаты или операторы языка и требовать их запоминание, а показывать, как находить задачи из реального мира, представлять их в виде информационных систем полезные человеку, затем подсказать какие из предикатов или операторов языка более подходящие для реализации этих систем. Начинать программировать с того языка, который наиболее доступный для понимания. И тогда нельзя не заметить, что правильнее будет начинать программирование с языка Пролог [2] (или ему подобному языку), хотя автор данной статьи предостерегает читателя, на этот счёт есть и много иных мнений. Но уж очень подходит и для гуманитариев, и для реалистов. Более привлекателен тем, что легче и быстрее его понять, а значит и полюбить. И это не только язык программирования, но и мощное средство для развития логического мышления для того, кто на нём программирует. Но это ещё мощное средство для моделирования. Логика высказываний, логика предикатов – весьма приемлемые средства для обработки естественного языка. А теория языка Пролога базируется на этих логиках.

Итак, начать обучение с декларативных языков программирования (Пролог, LISP, и др.) представляется более логичным. Например, если начинать с таких как язык логического программирования Пролог, шансы на успех в его изучении будут выше. Почему?

Во-первых, здесь необходимо знать на начальном этапе, **что** представляет собой проблема и ожидаемый результат, а **как** решает задачу интерпретатор можно иметь пока только общее представление.

Во-вторых, процесс программирования и интерпретации можно продемонстрировать весьма просто и довольно убедительно на примере написания и чтения текста естественного языка. Автор текста на естественном, например румынском языке, пишет его (текст) сверху вниз, слева направо, предложение за предложением. Этими предложениями описываются отношения между объектами (факты) или вывод новых фактов из уже написанных выше в тексте отношений (правила). Так автором текста описывается модель реального мира, представленного в на естественном языке. Читатель обращается к написанному автором тексту, тогда когда у него возникает цель (вопрос) познать что-то, полагая, что в этом тексте он может найти ответ на свой вопрос. Читает

этот текст также как писал его автор, сверху вниз, слева направо, пока не находит ответ на возникший у него вопрос, или не убеждается что в этом тексте, необходимого ответа на его вопрос нет. Более того, вопрос может касаться необходимости получения в ответе полной картины мира (смоделированной в конкретном тексте) или только её части (какой-то конкретный факт, отражающий только часть этого мира).

Текст, написанный на естественном языке автором, в котором логически излагается что-то из реального мира, можно сравнить с компьютерной программой написанной программистом-прикладником (тем же учеником) на языке логического программирования Пролог. А в качестве «читателя» программы выступает другая компьютерная программа (интерпретатор), один раз написанная системным программистом (разработчиком собственно языка Пролог) и поставляемая программистам-прикладникам для многократного применения. Принцип работы интерпретатора на первых порах можно продемонстрировать действиями читателя текста естественного языка. Мотивированный конкретным вопросом (целью), читатель сравнивает вопрос с фактом в тексте, сверху вниз, слева направо. Найдя подходящий факт, выводит ответ себе (читателю, пользователю). Более подробно о роли языка Пролог в изучении курса информатики было изложено в [2].

### **Выводы**

Проанализировав состояние с изучением информатики в школе (гимназия, лицей) вполне резонно прийти к следующим выводам:

1. Школьные программы курса информатики и соответствующие учебники, учитывая динамичность информатики как научного направления, весьма скромные успехи в освоении курса информатики в школе, должны публично анализироваться, оцениваться их результативность (уровень и достаточность полученных знаний выпускников лицеев) и пересматриваться в случае необходимости, практически ежегодно. Было бы весьма полезным, если бы структуры, отвечающие за школьное образование ещё более тщательно, организовывали бы публичное обсуждение учебных программ, учебников, пособий для учителей и т. д., чтобы никто из считающих себя авторитетом от науки и образования, не имел возможность каким-либо образом воспрепятствовать этому. Необходимо находить и устранять (в случае выявления) любые лазейки для монополизации деятельности, связанной с написанием учебников и других пособий, устранять препятствия для утверждения лучших их вариантов. Учёных и педагогов в республике много, не может быть, чтобы все были настолько пассивны. Много лет подряд, только единицы думали и писали лучше других. А если и так, то результаты должны постоянно подтверждать это. Ротация необходима для получения лучшего качества.
2. Для стимулирования возможностей появления новых научно-методологических подходов по содержанию учебных программ, использовании ТСО и методологии преподавания курса информатики в школе, курсов для будущих учителей информатики, таких как методы программирования и др., не практиковать официальное оппонирование научных работ на соискание ученых степеней молодых учёных со стороны лиц, заинтересованных в сохранении без изменений учебных программ в течении многих лет. Исключить в принципе возможность наличия в явном или неявном виде так называемых «штатных оппонентов» при любых научных советах (если такие возможности вдруг, где-то могут появиться).
3. Было бы полезным для дела, если бы ученые, педагоги и специалисты подвергли бы серьёзной, основательной критике приведенное здесь видение о примерном содержании учебной программы по изучению информатики в школе. Привели бы убедительные аргументы в пользу сохранения и защиты утверждённой и действующей программы или высказали бы свои мотивации необходимости её замены, выдвинули бы свои идеи по содержанию новой программы. Полемика на эту тему, привела бы к действительно хорошему решению.

## Библиография

1. N. Pelin. Studiarea programării logice în Moldova. Analele științifice ale doctoranților și competitorilor ”Probleme actuale ale științelor umanistice”. Vol. 1, Ch.:, 1996.
2. Н. Пелин. Роль логического программирования в изучении информатики. Revista metodică-științifică trimestrială „Administrarea Publică”, nr.3, 2016.
3. Специализированный сайт по логическому программированию и проектированию информационных систем [www.prolog.md](http://www.prolog.md).
4. А. Гремальски, Г. Василаке, Л. Гремальски. Информатика: Учеб. для 7 кл. – Ch.: Î.E.P. Știința, 2005.
5. А. Гремальски, Л. Гремальски. Информатика: Учеб. для 8 кл.–Ch.: Î.E.P. Știința, 2005.
6. А. Гремальски, И. Чобану. Информатика: 8 кл.: Пособие для учителя. – Ch.: Î.E.P. Știința, 2005.
7. А. Гремальски, Ю. Мокану, И. Спинеи. Информатика: Учеб. для 9 кл. – Ch.: Î.E.P. Știința, 2006.
8. A.Gremalschi, Informatică: Cl. a 9-a: Ghidul profesorului. – Ch.: Î.E.P. Știința, 2006.
9. А. Гремальски, Ю. Мокану, Л. Гремальски. Информатика: Учеб. для 10 кл. – Ch.: Î.E.P. Știința, 2007.
10. А. Гремальски. Информатика: Методы программирования: Учеб. для 11 кл. – Ch.: Î.E.P. Știința, 2005.
11. A.Gremalschi, S. Corlat, A. Braicov. Informatică: Cl. a 12-a. – Ch.: Î.E.P. Știința, 2010.
12. DEX. Ediția a II-a. – В.: Univers enciclopedic, 1996.
13. Википедия. <https://ru.wikipedia.org/wiki/Информатика>
14. БСЭ том 10 изд.3-е. М.: «Советская энциклопедия», 1972.
15. Dicționar de informatică. Coordonare: Pierre Morvan.Traducere din franceza. – В.: Editura Niculescu, 2000.
16. [https:// ru.wikipedia.org/wiki/Язык\\_программирования](https://ru.wikipedia.org/wiki/Язык_программирования).
17. [https:// ru.wikipedia.org/wiki/Декларативное\\_программирование](https://ru.wikipedia.org/wiki/Декларативное_программирование).