

## ДВАДЦАТЬ ЛЕТ СПУСТЯ

*Галина Михайловна Рудакова, к.ф.-м.н., профессор*

*Тел.: 8 983 150 75 29, e-mail: gmrfait@gmail.com*

*Сибирский государственный технологический университет*

*http://www.sibstu.kts.ru*

*Сергей Петрович Якимов, к.т.н., доцент*

*Тел.: 8 391 265 30 01, e-mail: ysp2005@yandex.ru*

*Сибирский государственный технологический университет*

*http://www.sibstu.kts.ru*

*В статье речь идёт об обучении программированию в вузе, с использованием подхода от литературного текста к программному коду. Надо читать классику и видеть в ней задачу, уметь анализировать текст как документ. Авторы делятся своим двадцатилетним опытом работы в этом направлении.*

*Ключевые слова программирование, информационные технологии, языки высокого уровня*

Информационные технологии уже много лет возглавляют список специальностей, наиболее востребованных практикой. Требования к IT-специалисту, особенно занима-



**Г.М. Рудакова**

ющемся разработкой программных продуктов, возрастают с каждым днем. При этом, требования в значительной степени затрагивают аспекты, связанные с умением быстро адаптироваться не только к смене инструментария, но и к предметной области.

Вспоминая известную поговорку, можно утверждать, что программист должен быть готов к ситуации, когда он «и швец, и жнец, и на дуде



**С.П. Якимов**

игрец». Сегодня он пишет программу для врача, завтра – для сталевара, при этом ухитряется перейти на новую техническую базу, использующую принципиально новый инструментарий. Меняется всё, базовым остается умение структурно мыслить и программировать, являющееся непременным условием глубокого освоения современных информационных технологий и новых, ранее не знакомых предметных областей. И именно этот навык в первую очередь необходим будущему «айтишнику». Наличие таких специалистов и качество их подготовки во многом определяют темпы развития общества.

Обучение программированию (абсолютно не принципиально, какой язык изучается, – всё равно, как показывает опыт, рано или поздно инструментарий придется менять, и менять многократно) традиционно, проходит по сценарию: формализованный текст задачи, построение алгоритма в виде блок-схемы, написание программы на языке, тестирование, результат. Заметим, в этой схеме отсутствуют такие немаловажные аспекты практической деятельности программиста, как нечеткость постановки задачи, поиск и обсуждение алгоритмически оригинального решения, поиск новых математических подходов, и, самое главное, практически отсутствует возможность отстоять свою точку зрения на решение и верность самой постановки задачи из-за ее жесткой заформализованности. Ситуация усугубляется тем, что современные студенты, даже учащиеся в одной группе, плохо знают друг друга: социальные сети, телекоммуникационное общение в значительной степени подменяют им «щедрость человеческого общения», подчас доводят молодого человека до болезненного состояния [1]. При таком

подходе не приходится говорить о преемственности поколений.

В отличие от традиционной технологии обучения ИТ-специалистов мы двадцать лет назад применили подход «от литературы к программированию», правда не на обычных лабораторно-практических занятиях, что даже сейчас из-за дефицита дидактического материала представляется затруднительным, а при проведении региональных олимпиад по программированию среди студентов с 1992 по 2002 гг. В отличие от принятого в ИСМ, Положением о региональной олимпиаде по программированию были предусмотрены такие этапы, как теоретический тур, апелляция, практический тур, обсуждение наилучших решений, разбор типичных ошибок.

Стремительно развиваются информационные технологии, появляются новые пакеты и компоненты, позволяющие разрабатывать сложнейшие системы, но по-прежнему не снижается потребность в ИТ-специалистах, умеющих воспринимать плохо формализованную информацию, увидеть в ней задачу, формализовать ее, алгоритмизировать и, в конечном итоге, довести до программной реализации. И как всегда, «новое – это хорошо забытое старое». Может быть, такую технологию, которую мы использовали двадцать лет назад, стоит использовать сейчас в школе, в вузе при обучении программированию, особенно на фоне затянувшегося кризиса образования [2; 3]. Ведь анализ литературного текста позволяет посмотреть на него как на документ, по которому можно построить алгоритм всех действий – от написания технического задания до кодирования [4].

В данной статье мы приводим малую долю наших задач теоретического и практического туров. Таких «литературных» заданий у нас за 10 лет проведения олимпиад с 1992 г. по 2002 г., было придумано-создано более сотни. На теоретическом (первой части соревнования) туре на составление программы без ЭВМ давалось 4 часа. Каждый участник работал самостоятельно и буквально писал ручкой. Текст программ(ы) писался на бланках, выданных жюри, разборчиво, сопровождался подробными комментариями, должен был отвечать требованиям и стандартам структурного программирования. Затем, на апелляции могли присутствовать желающие участники и обязательно все члены жюри. Этап апелляции давал реальную возможность отстоять свою интерпретацию постановки задачи и предложенное решение. Далекое не всегда окончательная оценка, поставленная жюри, совпадала с первоначальной. Оценивался теоретический тур: запись программы решения задачи на языке высокого уровня до 100% оценки задачи, частное решение до 50 % от оценки задачи. На практическом туре баллы присуждались только за программы, прошедшие тестовый контроль. На этом этапе уже проверялись в первую очередь практические навыки быстрого решения формализованных задач.

### **Задания теоретического тура**

#### **1. Комар (200-летию А.С. Пушкина посвящается)**

*... Гости в путь, а князь Гвидон*

*С берега душой печальной*

*Провожает бег их дальный;*

*Глядь – поверх текучих вод*

*Лебедь белая плывет.*

*“Здравствуй, князь ты мой прекрасный!*

*Что ты тих, как день ненастный?*

*Опечалился чему?” –*

*Говорит она ему.*

*Князь печально отвечает:*

*“Грусть-тоска меня съедает,*

*Одолела молодца;*

*Видеть я б хотел отца”.*

*Тут он в точку уменьшился,*

*Комаром оборотился,*

*Полетел и запищал,*

*Судно на море догнал,*

*Потихоньку опустился*

*На корабль – и в щель забился.*

*Ветер весело шумит,*

*Судно весело бежит*

*Мимо острова Буяна,*

*К царству славного Салтана,*

*И желанная страна*

*Вот уж издали видна.*

*Вот на берег вышли гости;*

*Лебедь князю: “Вот в чем горе!  
Ну, послушай: хочешь в море  
Полететь за кораблем?  
Будь же, князь, ты комаром”.  
И крылами замахала,  
Воду с шумом расплескала  
И обрызгала его  
С головы до ног – всего.*

*Царь Салтан зовет их в гости,  
И за ними во дворец  
Полетел наш удалец.  
Видит: весь сияя в злате,  
Царь Салтан сидит в палате  
На престоле и в венце,  
С грустной думой на лице...*

А вокруг, как и полагается – толпа из  $N$  придворных, найти в которой ткачиху с поварихой, с сватьей бабой Бабарихой не так-то просто. Комар (князь Гвидон), поразмыслив, расположился в палатах таким образом, что в зону его видимости радиусом  $R$  попало максимально большее количество народу. Определить координаты хотя бы одной такой точки в царских палатах, где мог расположиться комар для наблюдения за ходом политической интриги, в результате которой

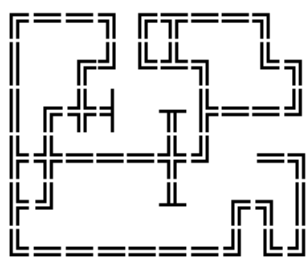
*Повариха побледнела,  
Обмерла и окривела,*

если известны координаты  $(x_i, y_i)$  всех придворных, находившихся при этом в зале.

*Максимальная оценка: 30 баллов.*

## 2. Неизвестная история доктора Джеккила и доктора Хайда (по Стивенсону)

Среди множества приключений доктора Джеккила и доктора Хайда есть одно, не



попавшее в поле зрения Стивенсона. Им стало известно о существовании древнего клада в одном из коридоров культовой пещеры Гдетотам. Независимо друг от друга они приступили к поискам. Оказалось, что некоторые коридоры древними строителями были перегородены кирпичной кладкой, поэтому каждому кладоискателю пришлось запастись киркой. С помощью кирки, конечно, можно разбить стенку, но при этом тот, кто это делает, теряет единицу энергии, правда не совсем,

так как энергия другого ровно на столько же при этом возрастает. Изначально и д-р Джеккил и д-р Хайд обладают порцией энергии в  $E$  единиц каждый. Читавшим их историю в изложении Стивенсона не трудно догадаться, что если только хотя бы у одного из кладоискателей энергия падает до нуля – гибнут оба. Обладателем клада становится тот, кто первым его обнаружит. Необходимо найти вариант поисков, при котором сокровища будут вынесены за пределы лабиринта, в предположении, что каждый из кладоискателей обладает двумя степенями свободы.

План пещеры и начальное положение д-ра Джеккила и д-ра Хайда заданы в символьном файле `gdetotam.dat`. При этом используются следующие обозначения:

Символ	Код ASCII	Назначение
Пробел	#32	Коридор пещеры
W	#87	Стена пещеры
К	#175	Кирпичная кладка
*	#42	Клад
Enter	#13	Переход к следующей дискретной строке плана пещеры
В	#66	Вход в пещеру или выход
J	#74	Положение в пещере доктора Джеккила
Н	#72	Положение в пещере доктора Хайда
А	#65	Положение в пещере д-ра Джеккила и д-ра Хайда в том случае, когда они находятся в одной точке

*Максимальная оценка: 50 баллов.*

### 3. А роза упала на лапу азора

В исходную строку (не более 255 символов) записано несколько слов. Разрешается многократно и в любых комбинациях выполнять следующие операции:

- а) менять местами два любых слова,
- б) удалять из слова одну букву, получая при этом новое слово.

Получить фразу-перевертыш максимально возможной длины. При подсчете длины фразы разделители между словами игнорируются.

*Максимальная оценка: 15 баллов.*

### 4. Монетный двор на Кин-дза-дза

Двум эцилопам – Мряке и Бряке захотелось независимости чеклан от голодных хрюмзиков. С этой целью они решили чеканить собственную золотую и серебряную монету. С золотой монетой проблем не возникло: начали чеканить 1 каце (в одной каце –  $N$  цаков). Пресс-форму для каце Мряка схряпала вместе с гравицапой от пепелаца у Горбика, пока тот находился в эцихе за то, что щелкал клювом на Борика. А вот с разменной монетой все оказалось не так просто. Так как ни Мряка, ни Бряка не умели сами изготавливать пресс-формы, то их пришлось заказывать у жадного Слюника, а тот, будучи махровым кю, заломил за них очень большую цену в конвертируемой валюте. В то же время, проклятые кю отказались предоставлять чекланам очередной транш кю.

Между Мрякой и Брякой разгорелись споры по поводу того, как решить возникшую проблему. Мряка, следуя наказам обманутых ею избирателей, настаивала на разработке пресс-форм таким образом, чтобы можно было разменять любую сумму от одного цака до каце включительно. Бряка, учитывая эстетические запросы заживевших олигархов, которых мелкие суммы не интересовали, требовал чеканить монеты так, что, если взять по одной серебряной монете различного достоинства, то в сумме получалась бы каце. В результате Мряка облила соком Бряку на глазах у тысяч избирателей, а Бряка отгаскал Мряку за космы прямо в зале заседаний чекланского бредламента.

Попробуйте найти такой способ чеканки разменной монеты, который обеспечивал бы мирное сосуществование Мряки и Бряки, а также учитывал незавидное финансовое положение молодого независимого от голодных хрюмзиков государства чеклан.

*Оценки: Частное решение при  $N=100$ : 10 баллов.*

*Общее решение для произвольного  $N$ : 25 баллов.*

### 5. Шахматы

На шахматной доске выставлены 4 фигуры: 2 белые и 2 черные. За минимальное количество ходов поменять фигуры местами: каждая белая фигура должна встать на место изначально занятое какой-либо черной и наоборот. Начинают белые. Ходы делаются по очереди. Становиться на битое поле нельзя. Для печати ответа использовать стандартную шахматную нотацию. Например, ход конем с поля  $a1$  на поле  $b3$  записывается как  $Ka1 - b3$ . Обозначения:  $K$  – конь,  $C$  – слон,  $L$  – ладья,  $\Phi$  – ферзь. Начальное положение фигур и их названия вводятся с клавиатуры.

*Максимальная оценка: 15 баллов.*

### 6. Чечня

В результате бомбардировки федеральными войсками одного из кварталов города Энска были разрушены подземные водопроводные коммуникации. Задан план квартала, разбитого на квадраты. В каждом квадрате имеется не более одного фрагмента трубопровода. Восстановить схему коммуникации, если известно, что в результате взрывов все фрагменты остались в своем квадрате (изменилась только их ориентация на плоскости), и вся система водоснабжения жилого квартала имела выходы всего в двух точках. Данные о расположении фрагментов трубопровода записаны в текстовый файл *sun.dat* в виде матрицы  $A_{ij}$  размером  $N \times M$ , где  $A_{ij}$  – символ псевдографики, изображающий фрагмент, находящийся в квадрате с координатами  $i$  и  $j$ . При этом используются

следующие обозначения:  $\perp$ ,  $\dashv$ ,  $\perp\!\!\!\perp$ ,  $\perp\!\!\!\perp$  – заглушка;  $=$ ,  $\parallel$  – труба,  $\llcorner$ ,  $\lrcorner$ ,  $\ulcorner$ ,  $\urcorner$  – уголок,  $\overline{\perp\!\!\!\perp}$ ,  $\underline{\perp\!\!\!\perp}$ ,  $\overline{\perp}$ ,  $\underline{\perp}$  – тройник,  $\overline{\perp\!\!\!\perp}$  – пересечение двух труб. Результаты расчетов вывести символами псевдографики на экран в виде плана трубопровода, как это показано на рисунке.

*Максимальная оценка: 45 баллов.*

### 7. Бесконечная дробь

Дано:  $n, a, x$  – натуральные числа, не превышающие 100. Вычислить с точностью до периода значение функции:

$$Y = \frac{n!(x^2 n! + 2x^3 - n!a^2 - 2a^2 x) - 2(n! + x) + x^4 - (xa)^2}{xn! - an! + x^2 - ax} + \frac{2}{n!x - an! + x^2 - ax}$$

*Максимальная оценка: 5 баллов.*

### Задания практического тура

#### 1. Греческая смоковница

Положение  $N$  городов задано парами координат. Между некоторыми из них имеются скоростные прямолинейные автострады с двусторонним движением. Кроме этого из каждого города в ближайший населенный пункт, с которым еще нет сообщения по прямой линии, начинает строиться одна новая прямолинейная автострада. Скорость прокладки дороги  $V_d$ . Известно, что среди городов нет трех, находящихся на одной прямой, и все расстояния между городами различны. Написать программу, рассчитывающую маршрут студентки, путешествующей автостопом, позволяющий ей попасть из  $i$ -го города в  $j$ -й за кратчайшее время. Исходить следует из предположения, что все новые дороги начали строиться одновременно и не позже, чем путешественница выехала из  $i$ -го города. Отсчет времени путешествия ведется от момента начала строительства дорог. Скорость движения по автостраде  $V_a$ . Исходная сеть дорог задана матрицей смежности

$$C_{k,l} = \begin{cases} 1 & \text{если непосредственно из города } k \text{ в город } l \text{ (} k \neq l \text{) есть дорога} \\ 0 & \text{в оставшихся случаях} \end{cases}$$

Данные вводятся из текстового файла. Имя файла с исходными данными вводится с клавиатуры. Описание формата файла исходных данных:

*<скорость движения>*

*<скорость строительства автострады>*

*<количество городов>*

*<номер начального пункта путешествия>: <номер конечного пункта путешествия>*

*<номер первого города>: <первая координата>; <вторая координата>*

...

*<номер N-го города>: <первая координата>; <вторая координата>*

*<матрица смежности построено>*

Требования к представлению результатов: перечислить пункты следования, включая начальную и конечную точки. Указать время, затраченное на путешествие.

Тест для отладки находится в Вашем каталоге в файле *test\_1.dat*.

*Максимальная оценка: 50 баллов (2 теста по 25 баллов).*

*Каждый тест имеет ограничение по времени 5 минут.*

#### 2. На берегах Онтарио

На экране дисплея должна быть представлена карта равнинной местности. На ней в виде многоугольников изображается единственное озеро и несколько островов на

нем. Путешественник сидит в лодке, которая находится на водной поверхности в точке  $A$  с координатами  $(X_A, Y_A)$ . Проложить по карте кратчайший путь, по которому путешественник может попасть в точку  $B(X_B, Y_B)$ . Контурь озера и островов заданы с помощью координат вершин всех своих углов в координатах дисплея. Пары координат в порядке обхода периметра контура по часовой стрелке записаны через пробелы в текстовый файл, в котором каждая строка определяет один контур. Первый контур задает береговую линию озера. Полное имя файла данных и координаты начальной и конечной точки маршрута вводятся с клавиатуры. Тест для отладки находится в Вашем каталоге в файле *test\_2.dat*.

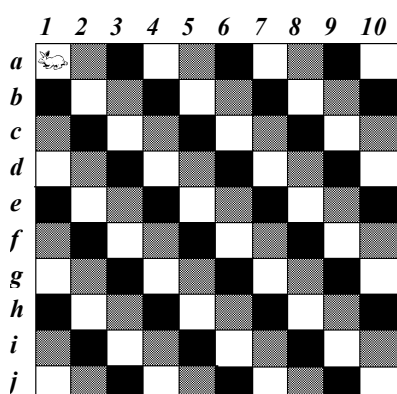
P.S. Варианты, при которых лодка перемещается посуху волоком или иным жульническим способом, жюри принимать во внимание не будет, ссылки на исторические, литературные и прочие прецеденты не принимаются.

*Максимальная оценка: 40 баллов (4 теста по 10 баллов).*

*Каждый тест имеет ограничение по времени 5 минут.*

#### 4. Тау-Китайские шахматы

<i>Покамест я в анабиозе лежу,</i>	<i>У Таукитов</i>
<i>Те таукитяне буянят, –</i>	<i>В алфавите слов</i>
<i>Все реже я с ними на связь выхожу:</i>	<i>Немного и строй – буржуазный,</i>
<i>Уж очень они хулиганят.</i>	<i>И юмор у них – безобразный.</i>



Мало им было трехмерных шахмат, которые они представили на 1-ю олимпиаду. На этот раз в три цвета раскрасили доску (как показано на рисунке) и придумали динамическую фигуру, которая может превращаться в коня, кролика или дракона. Двигаясь по шахматной доске, она превращается в кролика, если попадает на белую клетку, в дракона, попадая на серую, и в коня – на черную. Конь ходит как обычный шахматный конь, кролик сдвигается на одну клетку по горизонтали или вертикали, а дракон летает на четыре клетки по диагонали.

Дана шахматная доска  $N \times N$  клеток, начальное положение фигуры – левый верхний угол (a1), окрашенный в белый цвет. Необходимо найти для нее кратчайший путь к произвольной заданной клетке доски.

Все исходные данные: размер доски и координаты конечной клетки, вводятся с клавиатуры. Требования к представлению результатов:

1) изобразить шахматную доску; 2) отразить на изображении доски маршрут; 3) вывести сообщение о количестве сделанных фигурой ходов.

*Максимальная оценка: 20 баллов (4 теста по 5 баллов).*

*Каждый тест имеет ограничение по времени 5 минут.*

*В случае непрохождения хотя бы одного теста, задача не засчитывается.*

#### 5. Бармаглот (Алиса в зазеркалье. Льюис Кэрролл)

<i>Воркалось. Хливкие шорьки</i>	<i>О, бойся Бармаглота, сын!</i>
<i>Пырялись по наве,</i>	<i>Он так свиреп и дик,</i>
<i>И хрюкотали зелюки,</i>	<i>А в глуши рымит исполин –</i>
<i>Как мюмзики в мове.</i>	<i>Злопастный Брандашмыг!</i>

Русскоязычный текст записан в текстовый файл. Из этого текста взят какой-то фрагмент и зашифрован путем замены каждой буквы одно- или двузначным десятичным числом, взаимно однозначно ее идентифицирующим и записан в другой текстовый файл. Имена файлов вводятся с клавиатуры. В качестве разделителей между словами

используется пробел (один или несколько) или перевод строки (один или несколько). Все знаки препинания (в т.ч. дефис), за исключением тех, которые стоят в конце предложения при шифровке заменяются пробелом. Если знак препинания находится в конце предложения, то он заменяется точкой. Буквы «е» и «ё» шифруются одинаково. Например, если в качестве шифра используется порядковый номер буквы в алфавите, цитата из «Алисы в зазеркалье», вынесенная в эпиграф задачи шифруется следующим образом:

```
3151711112151827. 2212931196 25151727119 162817321291827 1615 14136 9
22173111115191129 861231119 11111 13311389119 3 131536. 15 215101832
211713141215191 182814. 1514 19111 183917616 9 5911 1 3 41220259 172813919
918161512914 812151611819142810 217114512513284.
```

Найти в тексте зашифрованную фразу, вывести в файл с произвольным именем ее и восстановленную часть ключа.

Текст для отладки находится в Вашем каталоге в файлах *alisa.asc* и *alisa.ans*. Зашифрованная фраза для отладки по этому тексту – в файле *f0.txt*.

*Максимальная оценка: 30 баллов (3 теста по 10 баллов).*

*Каждый тест имеет ограничение по времени 5 минут.*

*В случае непрохождения первого теста, следующие тесты не проверяются.*

### **5. Всемирный потоп**

Высокогорный массив в районе горы Арарат имеет в плане прямоугольную форму и задан массивом действительных чисел  $X[i, j]$ , характеризующих высоту точки с целочисленными координатами  $(i, j)$ ,  $i = 1 \dots N, j = 1 \dots M$  над уровнем моря. В результате того, что в свое время хляби небесные разверзлись, горный массив был полностью затоплен. По окончании этого мокрого дела вода схлынула до прежнего уровня, но география местности вследствие того, что образовались озера, существенно изменилась. Подсчитать количество озер внутри заданного горного массива, вычислить их максимальную глубину и площадь водной поверхности. Исходить из предположения, что вода может переливаться только вдоль координатных осей.

Данные вводятся из текстового файла. Имя файла с исходными данными вводится с клавиатуры. Описание формата файла исходных данных:

*<значение N>*

*<значение M>*

*<матрица высот построчно через пробел>*

Отладочный пример представляет собой матрицу размером 10x10, записанную в файл *test\_5.dat*, который находится в Вашем каталоге:

9	9	9	9	9	9	9	9	9	9
9	3	3	3	3	3	3	3	3	9
9	2	2	2	2	2	2	2	2	9
9	1	1	1	1	1	1	1	1	9
9	2	2	2	2	2	2	2	2	5
9	3	3	3	3	3	3	3	3	9
9	4	4	4	4	4	4	4	4	9
9	5	5	5	5	5	5	5	5	9
9	2	2	2	2	2	2	2	2	9
9	9	9	9	9	9	9	9	9	9

В результате обработки теста должно получиться: 2 озера, максимальная глубина 4, площадь водной поверхности 56

*Максимальная оценка: 30 баллов (5 тестов по 10 баллов).*

*Каждый тест имеет ограничение по времени 5 минут*

Вот любопытнейшие высказывания на одном из форумов [5] (орфография сохранена):

1 «Я просто не представляю всех, кто учатся в вузах на программистов. Как они устроятся потом на работу? Ведь там нечему не учат!

Программирование это искусство, которому нельзя научиться по книге, это бесконечная наука, где нужно решать тысяча головоломок, чтобы вложить знания в каких-то 5 строчек, в вузах же просто сидят и пишут синтаксис и примеры по сложности 2+2».

2 «Я вот программист самоучка. Ходил, разве что на "кружок программирования basic" на еще древних машинах zx-spectrum, мож кто слышал,... потом сам выучил assembler под него, сейчас вот уже 4 года шпарю js/php, недавно начал поднимать C++/Direct3D... Если голова варит и есть малейшее понятие, как тебе это изучить, то гуглишь, ищешь книги и тупо изучаешь с практикой... Главное понять схему изучения того или иного материала.

А этому учат в институтах – работать с литературой!».

Вывод напрашивается сам: нельзя гоняться за новыми языками – учить надо мышлению, а не синтаксису.

### **Литература**

1. *Котляров А.В.* Другие наркотики или Homo Addictus: Человек зависимый. – М.: Психотерапия, 2006. 480 с.

2. *Марфенин Н.Н.* Чему и как учить для устойчивого развития? // Россия в окружающем мире: 2010. Устойчивое развитие: экология, политика, экономика: Аналитический ежегодник. – М.: МНЭПУ, 2010. С. 146-177.

3. *Якимов С.П.* Функциональная модель процесса образования. // Открытое образование. 2012. № 5. С. 48-54

4. *Рудакова Г.М.* Причины и движущие силы поэтапной смены подходов внутри объектно-ориентированной парадигмы программирования // Образовательные ресурсы и технологии. 2014. №1(4). С.68-78

5. <http://javascript.ru/forum/offtopic/6749-programmirovaniyu-nelzya-nauchitsya-v-vuzakh-ehto-iskusstvo-kotoromu-uchatsya-sami.html>

### **Twenty years later**

*Galina Mikhailovna Rudakova, PhD, Professor*

*Sergey Petrovich Yakimov, Associate Professor*

*In the article we are talking about learning programming in high school using technology from the literary text to code. One must read classics and understand its meaning, be able to analyze the text as a document. The authors share their 20 years of experience in this direction.*

*Keywords – programming, information technologies, high-level languages*

УДК 004.89

## **ДИСТАНЦИОННОЕ ОБУЧЕНИЕ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗРЕНИЯ**

*Галина Ахметовна Самигулина, д.т.н., зав. лабораторией*

*Интеллектуальные системы управления и прогнозирования*

*Тел.: 8 727 272 4617, e-mail: galinasamigulina@mail.ru*

*Институт информационных и вычислительных технологий КН МОН РК*

*<http://www.ipic.kz>*