

**Александр Дуванов  
Алексей Рудь  
Виктор Семенко**

# **АЗЫ ПРОГРАММИРОВАНИЯ**

**ФАКУЛЬТАТИВНЫЙ КУРС**

**ЗАДАЧНИК**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06(076.1)  
ББК 32.973я721  
Д79

**Дуванов А. А., Рудь А. В., Семенко В. П.**

Д79 Азы программирования. Факультативный курс. Задачник. — СПб.: БХВ-Петербург, 2005. — 160 с.: ил.

ISBN 5-94157-288-3

В книге собраны задачи, которые предлагались на сетевых турнирах курса «Азы программирования» Роботландского университета. Представлены задачи различной тематики: поиск и сортировка, нахождение экстремума, разнообразные подсчёты, логические и геометрические задачи; традиционный наибольший общий делитель и Ханойская башня; нетрадиционные для школьников рекурсивные определения Бэкуса-Наура, запись выражения по-польски, стековые вычисления, синтаксический анализ и построение трансляторов. Задачи ориентированы на программирование в среде исполнителей Кукарача и Корректор, подробное описание которых содержит книга «Азы программирования. Книга для ученика».

*Для учащихся средних образовательных учреждений*

УДК 681.3.06(076.1)  
ББК 32.973я721

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. гл. редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Михальчук</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн обложки	<i>Инны Тачиной</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.06.05.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 10.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию  
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой  
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-288-3

© Дуванов А. А., Рудь А. В., Семенко В. П., 2005  
© Дуванов А. А., Русс А. А., иллюстрации, 2005  
© Оформление, издательство "БХВ-Петербург", 2005

# Оглавление

<b>Состав комплекта .....</b>	<b>7</b>
<b>Вступление .....</b>	<b>9</b>
Исполнители и язык программирования .....	9
Исполнитель Кукарача.....	9
Исполнитель Корректор .....	11
<b>Организация обучения на курсе «Азы программирования» .....</b>	<b>15</b>
Сетевые турниры .....	15
Решение задач .....	16
Перекрестная проверка работ .....	16
Общие рекомендации по перекрёстной проверке .....	17
Этическая дисциплина проверки и её полнота .....	17
Общий критерий оценок .....	18
Апелляционный период .....	18
Подведение итогов.....	18
Говорят студенты курса «Азы программирования» .....	19
Маги и Магистры программирования .....	20
Список Бессмертных Магов (по состоянию на 2004 год) .....	20
История Кукарачи .....	21
История Корректора .....	22
<b>Глава 1. Олимпиада 1996/1997.....</b>	<b>23</b>
Турнир Кукарачи.....	23

<b>Глава 2. Турниры 1997/1998 .....</b>	<b>31</b>
2.1. Турнир Кукарачи .....	31
2.2. Турнир Корректора .....	38
<b>Глава 3. Турниры 1998/1999 .....</b>	<b>45</b>
3.1. Турнир Кукарачи .....	45
Пояснительная записка.....	45
Разминка .....	46
Синтаксический анализ записи .....	48
Кукарачья геометрия .....	51
3.2. Турнир Корректора .....	56
Синтаксический анализ выражений и трансляторы .....	56
Геометрия на ленте Корректора .....	61
<b>Глава 4. Турниры 1999/2000 .....</b>	<b>67</b>
4.1. Турнир Кукарачи .....	67
Разминка .....	67
Синтаксический анализ записи .....	70
Кукарачья геометрия .....	75
4.2. Турнир Корректора .....	78
Система команд Малыша .....	82
Алгоритм выполнения программы .....	83
Авост.....	83
Задания.....	84
<b>Глава 5. Турниры 2000/2001 .....</b>	<b>85</b>
5.1. Турнир Кукарачи .....	85
Построения на клетчатом поле .....	85
Анализаторы и трансляторы.....	90
5.2. Турнир Корректора .....	92
Соглашения .....	92
Рекуррентные последовательности.....	93
Лексический анализ выражений.....	96
Классические алгоритмы .....	96

---

<b>Глава 6. Турниры 2001/2002 .....</b>	<b>99</b>
6.1. Турнир Кукарачи .....	99
Вступительное слово куратора .....	99
6.2. Турнир Корректора .....	106
Вступительное слово куратора .....	106
<b>Глава 7. Турниры 2002/2003 .....</b>	<b>113</b>
7.1. Турнир Кукарачи .....	113
7.2. Турнир Корректора .....	117
Вступительное слово куратора .....	117
<b>Глава 8. Турниры 2003/2004 .....</b>	<b>125</b>
8.1. Турнир Кукарачи .....	125
8.2. Турнир Корректора .....	131
Определения .....	131
<b>Глава 9. Турниры 2004/2005 .....</b>	<b>139</b>
Олимпиада Кукарачи .....	139
<b>Ссылки на задачи.....</b>	<b>145</b>
Часть I. Кукарача.....	145
Часть II. Корректор.....	151
Часть III. Транслятор?.. Это очень просто .....	159

# Состав комплекта

Комплект «Азы программирования» содержит всё необходимое для построения факультатива, сопровождающего школьный курс «Азы информатики». Он включает в себя:

- книгу «Азы программирования. Магия для начинающих» — учебник;
- книгу «Азы программирования. Задачи роботландских турниров» — задачник;
- книгу «Азы программирования» — пособие для учителя;
- CD с программными средами и дополнительными материалами (сопровождает книгу учителя).

# Вступление

В этой книге собраны задачи, которые предлагались на сетевых турнирах курса «Азы программирования» Роботландского университета в течение девяти учебных лет, начиная с первой общей олимпиады 1996/1997 и далее до олимпиады первого семестра 2004/2005 учебного года.

Задачи ориентированы на программирование исполнителей Кукарача и Корректор.

Подробное описание этих исполнителей и языка программирования содержит книга «Азы программирования. Магия для начинающих». Далее приводится лишь краткое, справочное описание Кукарачи, Корректора и языка программирования этих исполнителей.

## Исполнители и язык программирования

### Исполнитель Кукарача

Кукарача — исполнитель, работающий на клетчатом поле (рис. 1).



Рис. 1. Клетчатое поле с исполнителем и кубиками

Кукарача может переползать из клетки в клетку вверх, вниз, влево, вправо, но не по диагонали. Выход исполнителя за пределы поля запрещён (отказ «Не могу!»). В каждой клетке может находиться кубик с нанесённым на его грань символом. Кукарача, перемещаясь по полю, может толкать один или несколько кубиков перед собой и «сбрасывать» кубики за пределы поля.

Некоторые кубики на поле могут иметь «скрытые» символы, обозначаемые знаком вопроса. Считается, что такие кубики расположены в клетке «символом вниз». Толкая скрытый кубик, Кукарача перемещает его в следующую клетку по ходу движения, переворачивает и видит надпись.

Исполнитель в состоянии прочитать символ на кубике независимо от того, была ли запись символа обычной или скрытой.

Система команд исполнителя (СКИ) включает пять команд (рис. 2, табл. 1).



Рис. 2. Система команд Кукарачи

Таблица 1

Команда Кукарачи	Как выполняется
ВПРАВО	Сместиться на одну клетку вправо
ВЛЕВО	Сместиться на одну клетку влево
ВВЕРХ	Сместиться на одну клетку вверх
ВНИЗ	Сместиться на одну клетку вниз
СТОЯТЬ	Пустая команда — исполнитель не выполняет никаких действий

Любая команда, кроме команды **стоять**, может привести к отказу «Не могу», если её выполнение уводит Кукарачу за пределы поля.

Кукарача способен выполнять в своей среде проверки, представленные в табл. 2 (запись <символ> обозначает шаблон, на месте которого может быть любой символ).



Таблица 2

Запись условия	Результат проверки
<символ>	<i>Истина</i> , если Кукарача толкнул кубик с указанным в условии символом, <i>ложь</i> в противном случае
НЕ <символ>	<i>Ложь</i> , если Кукарача толкнул кубик с указанным в условии символом, <i>истина</i> в противном случае
ПУСТО	<i>Истина</i> , если клетка, в которую сместился Кукарача, пуста, <i>ложь</i> в противном случае
НЕ ПУСТО	<i>Ложь</i> , если клетка, в которую сместился Кукарача, пуста, <i>истина</i> в противном случае
ЦИФРА	<i>Истина</i> , если Кукарача толкнул кубик с цифрой (один из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9), <i>ложь</i> в противном случае
НЕ ЦИФРА	<i>Ложь</i> , если Кукарача толкнул кубик с цифрой (один из символов 0, 1, 2, 3, 4, 5, 6, 7, 8, 9), <i>истина</i> в противном случае

## Исполнитель Корректор

Корректор работает с длинной лентой, которая разбита на клетки (ячейки).

В ячейку может быть записан один символ. Символ — это буква, цифра, другие знаки, составляющие фиксированный алфавит исполнителя.

Считается, что лента спрятана в непрозрачный для Корректора футляр. Исполнитель видит только одну ячейку через специальное окно в футляре, и может записывать в «оконную» ячейку (окно ленты) любые символы из своего алфавита. Корректор может перемещать окно вправо и влево вдоль ленты и, таким образом, записывать символы в любые ячейки на ней.

В среде исполнителя есть ящик — специальная ячейка памяти. В ящик Корректор может снимать копию символа из окна и, наоборот, копировать символ из ящика в окно на ленту (рис. 3).



Рис. 3. Среда Корректора

На рис. 4 и в табл. 3 приводится полный набор команд Корректора и его алфавит.



Рис. 4. Система команд Корректора

Таблица 3

Команда Корректора	Как выполняется
ВПРАВО	Переместить окно на одну клетку вправо
ВЛЕВО	Переместить окно на одну клетку влево
ПИШИ <символ>	Записать указанный символ в клетку на ленте. В качестве символа можно указывать ключевые слова ПУСТО и ПРОБЕЛ
ЯЩИК+	Копировать символ с ленты в ящик
ЯЩИК-	Копировать символ из ящика на ленту
ОБМЕН	Поменять местами содержимое ящика и окна ленты
ПЛЮС	Заменить символ в окне символом, следующим по порядку в алфавите Корректора. Команда приводит к отказу (ситуация «Не могу»), когда в окне перед её выполнением записан последний символ алфавита
МИНУС	Заменить символ в окне символом, предыдущим по порядку в алфавите Корректора. Команда приводит к отказу (ситуация «Не могу»), когда в окне записан первый символ алфавита (специальный символ ПУСТО)
СТОЯТЬ	Пустая команда; её выполнение не вызывает никаких изменений в среде Корректора

Корректор способен выполнять в своей среде следующие проверки (табл. 4).

Таблица 4

Запись условия	Результат проверки
<символ>	<i>Истина</i> , если символ в окне совпадает с символом, указанным в условии, <i>ложь</i> в противном случае. В качестве символа можно указывать ключевые слова ПУСТО, ПРОБЕЛ и ЦИФРА
я=л	<i>Истина</i> , если символ в ящике совпадает с символом в окне, <i>ложь</i> в противном случае
я#л	<i>Истина</i> , если символ в ящике не совпадает с символом в окне, <i>ложь</i> в противном случае
я>л	<i>Истина</i> , если символ в ящике имеет больший порядковый номер в алфавите Корректора по сравнению с номером символа в окне, <i>ложь</i> в противном случае
я<л	<i>Истина</i> , если символ в ящике имеет меньший порядковый номер в алфавите Корректора по сравнению с номером символа в окне, <i>ложь</i> в противном случае

Перед любым условием может быть написан ключевой модификатор **НЕ**, который меняет смысл условия на противоположный.

## Язык программирования исполнителей

### Разделитель слов

Разделителем слов в языке служит один или несколько пробелов, а также конец строки.

### Программа

Программа представляет собой последовательность процедур, записанных в любом порядке. Имя процедуры может быть использовано наравне с командой из СКИ исполнителя.

### Процедура

Описание процедуры имеет вид, представленный на рис. 5.

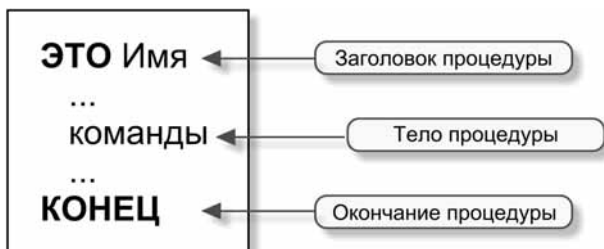


Рис. 5. Описание процедуры

## Имя процедуры

Имя процедуры должно начинаться с буквы. Оно не должно содержать пробелов и не может совпадать с ключевыми словами языка программирования.

## Комментарии

Комментарий имеет вид:

// текст комментария (до конца строки)

Комментарии могут располагаться как в процедурах, так и между ними.

## Команды

Таблица 5

Команда языка	Как выполняется
<команда из СКИ>	<b>Команда исполнителя</b> Выполнение определяется описанием команды в СКИ
<имя процедуры>	<b>Вызов процедуры</b> Выполняется процедура с указанным именем
ПОВТОРИ <число> <команда>	<b>Цикл ПОВТОРИ</b> Повторение выполнения команды указанное число раз
ПОКА <условие> <команда>	<b>Цикл ПОКА</b> Повторение выполнения команды, пока условие имеет значение <i>истина</i> . Проверка условия — перед выполнением команды
ЕСЛИ <условие> ТО <команда1> ИНАЧЕ <команда2>	<b>Команда ветвления</b> Проверяется условие и выполняется либо <команда1> (при значении условия: <i>истина</i> ), либо <команда2> (при значении условия: <i>ложь</i> ). Часть «ИНАЧЕ <команда2>» может быть опущена
{ <список команд> }	<b>Составная команда</b> Объединение нескольких команд в одну

# Организация обучения на курсе «Азы программирования»

Роботландский университет — это сетевая школа, в которой обучаются учителя и школьники индивидуально или совместно. Руководитель Роботландского университета — Дуванов Александр Александрович ([kurs@robotland.pereslavl.ru](mailto:kurs@robotland.pereslavl.ru)).

Во главе курса стоит куратор — сотрудник Роботландского университета.

В разное время кураторами «Азов программирования» были:

- Дуванов Александр Александрович, Переславль-Залесский (1996/1997 — 2000/2001);
- Рудь Алексей Владиславович, Снежинск (2001/2002, 2002/2003);
- Семенко Виктор Петрович, Рубцовск (2003/2004, 2004/2005).

Студенты курса образуют команды — группы детей, возглавляемые руководителем (как правило, школьным учителем).

В начале учебного года команды получают от куратора необходимые программно-методические материалы, календарные, тематические планы и приступают к работе.

## Сетевые турниры

В течение учебного года на курсе организуются два турнира — первый в конце первого семестра, второй — в конце второго.

Турнир состоит из четырёх этапов:

- решение задач;
- перекрёстная проверка;
- апелляционный период;
- подведение итогов.

## Решение задач

Условия задач (с оценочными баллами) публикуются на курсовой электронной конференции, и команды приступают к работе, на которую отводится две недели.

Задачи могут решаться коллективно, группами или индивидуально — турнирная тактика определяется командой (куратором никак не регламентируется).

В качестве результата команда высылает куратору одну общую работу, с обязательным указанием авторов решения каждой задачи.

## Перекрестная проверка работ

Проверка работ выполняется за одну неделю.

Все решения, присылаемые на конкурс, проверяются и рецензируются куратором, но кроме этого, сами ребята проверяют работы друг друга (по принципу команда из города А проверяет работы команд из городов В, С и D).

Перекрёстная проверка работ продолжает этап обучения. Она позволяет:

- Провести практические занятия с детьми по очень важному и ответственному этапу создания программного продукта — тестированию.

Ни одна компьютерная фирма не выпустит в свет свой программный продукт без тщательной и всесторонней проверки. Время, затраченное на тестирование продукта, заранее предусматривается, включается в план разработки и по количеству человеко-часов часто сравнимо со временем создания продукта. Тестируют программы специальные работники фирмы, как правило, не авторы программы. Как показывает опыт, автор программы не способен выявить в своём продукте большинство допущенных им ошибок. Объясняется это просто. Автор заставляет программу работать правильно и на это направляет свои усилия. Человек-тестер, наоборот, прилагает усилия для того, чтобы заставить программу работать неверно, «подсовывает» ей такие входные данные, на которых она начинает «хромать» — выдавать неверные результаты. Иными словами, у автора программы и проверяющего разные психологические установки: у одного — заставить программу работать, у другого — «сломать» её.

- Научить юных программистов читать чужую программу и понимать по её исходному тексту мысль автора.
- Познакомиться с решением других ребят и сравнить их со своими.

Перекрёстная проверка снабжается следующей инструкцией.

## Общие рекомендации по перекрёстной проверке

1. *Тестирование программы.* Первой, обязательной частью контроля, является проверка работоспособности программы на компьютере. Тестирование не должно сводиться к одному запуску программы с тем состоянием среды, которое предложено автором. Например, для Кукарачи запускайте программу с различными допустимыми (по условию задачи) положениями кубиков и исполнителя на поле. Обязательно рассмотрите критические (крайние) случаи и убедитесь, что программа обрабатывает их правильно.
2. *Проверка качества алгоритма.* Нужно внимательно прочитать объяснение алгоритма, приведённое автором, и проверить соответствие его программе при помощи запуска «по шагам». Если вы заметите в алгоритме погрешность, которая может привести к неверной работе программы, то придумайте тест (допустимое состояние среды), который обрабатывается программой неверно. Этот тест послужит доказательством наличия ошибки в программе.
3. *Оценка эффективности алгоритма и программы.* Оценивайте эффективность по трём признакам:
  - универсальность алгоритма (алгоритмы, которые работают не для всех возможных начальных состояний среды — ошибочные);
  - число строк программы с кодом (чем меньше, тем лучше);
  - время работы программы (чем меньше «движений» выполняет исполнитель в среде, тем лучше).
4. *Качество описаний алгоритма и внешний вид (рисунок) программы.* Изучая алгоритм автора, обратите внимание на качество описания, доступность и точность изложения, грамматическую правильность текста. Рассматривая программу, обращайтесь внимание на форматирование кода и оформление: комментарии, смысловые имена процедур, верно расставленные отступы и пустые строки улучшают внешний вид и способствуют лучшему пониманию программы.

## Этическая дисциплина проверки и её полнота

В инструкциях к перекрёстным проверкам куратор рекомендует командам помимо формальной оценки не скупиться на развернутые замечания и особенно на заслуженные комплименты.

Все замечания должны быть высказаны дружеским тоном, ведь оценивается не автор, а авторская работа. Работа может быть очень слабой (это разрешается!), но сам автор может быть очень талантливым, и его главный труд ещё впереди. Не надо считать недостатки работы недостатками автора.

Все оценки должны быть в обязательном порядке обоснованы (единым для всех сводом опубликованных критериев).

## Общий критерий оценок

1. Если задача не решалась командой, то за неё выставляется 0 баллов.
2. Если задача имеет несколько вариантов решения, то выставляется единственная оценка (балл) по лучшему варианту.
3. Программа не работает: выдается синтаксическая ошибка или сообщение «Не могу». Может, это недоразумение? Попробуйте обнаружить и исправить допущенную автором опisku.
  - 3.1. Описка. Если исправление помогло и программа заработала, снизьте оценку по рекомендациям для каждой отдельной задачи.
  - 3.2. Неверное решение. Если ошибки серьёзные и связаны с неверным алгоритмом или неверной программой, выставляйте за задачу 1 балл.
4. Программа работает, но в ней обнаружены грубые ошибки (например, программа работает только для того состояния среды, которое предложил автор) — выставляйте 2 балла.
5. Программа работает, но в ней есть ошибки (обнаружены состояния среды, в которых программа работает неправильно). Ставьте оценку в соответствии с приведёнными рекомендациями для каждой отдельной задачи.
6. В тех случаях, когда приведённые рекомендации оставляют неоднозначность, проверяющий принимает самостоятельное решение по выставлению оценки.

## Апелляционный период

Этот этап обучения длится одну неделю и совмещается с обсуждением решений, которое организует куратор.

Апелляции направляются на курсовую электронную конференцию и адресуются куратору или конкретным командам.

Студенты учатся вести дискуссию, публично отстаивая свою точку зрения. Ответ на апелляцию также имеет публичную форму.

В апелляции в корректной форме обосновывается претензия. Апелляции, основанные на эмоциях, не допускаются. Не допускаются упрёки к проверяющим, сравнение одних проверок с другими.

В апелляции указывается на конкретный пункт конкретной проверки и доказывается, что он противоречит конкурсным критериям оценок или что обозначенная в проверке ошибка отсутствует в работе.

## Подведение итогов

Куратор подводит итоги, публикуя в электронной конференции турнирную таблицу.



В этой таблице команды располагаются в порядке убывания итогового балла, вычисляемого как простая сумма набранных баллов по каждой задаче.

Балл за каждую задачу вычисляется по формуле:

$$\text{Балл} = ((\text{сумма } n \text{ оценок перекрестной проверки}) + 3 \cdot (\text{оценка куратора})) / (n + 3).$$

По результатам определяются победители: команды, занявшие I, II и III места.

В конце учебного года победителей ожидают:

- грамоты;
- дипломы Магов и Магистров;
- подарки от Роботландского университета.

## Говорят студенты курса «Азы программирования»

### Устич Дима

Мне понравился курс «Азы программирования». Сначала мы думали, что знаем всё о Кукараче, но оказалось, что знали совсем чуть-чуть. Но зато теперь мы можем решить задачу, используя рекурсивную пружину. А это совсем не просто. Но самое интересное, что нам было здорово вместе. Мы научились решать задачи сообща. Один предложит идею, а все вместе додумаем решение до конца.

### Галушко Марина

Мне понравилось дистанционное обучение. Но было очень трудно. Задачи были сложные, мы очень подробно их разбирали. Когда приходили конкурсные задачи, то забывали обо всём остальном. И решали, решали, решали. Дома, в школе, все вместе, по одному. Хотя и было трудно, но было интересно.

### Бердышева Юлия

Кажется, что эту задачу решить просто невозможно, а потом каждый говорит то, о чём думает, появляется интересная мысль и вырисовывается ход решения. Это очень здорово — решать трудные задачи вместе. А сколько у всех радости, когда задача уже решена, и Кукарача делает именно то, что мы хотим.

### Комлев Дима

Мне очень понравилось учиться в Роботландском университете, потому что мы не только научились решать задачи, но и увидели, как это делают

другие. Помню, нам очень понравились решения ребят из Снежинска. Мы очень тщательно тестировали их решения, а ошибок всё равно не было. Мы даже обнаружили ошибки в своих решениях, когда проверjali эту команду.

## Маги и Магистры программирования

**Маг** — это руководитель группы, которая когда-либо заняла в курсовом конкурсе первое место.

**Магистр** — это руководитель группы, которая в течение одного (любого) учебного года приняла участие в двух семестровых конкурсах и набрала на каждом не нулевую сумму баллов.

## Список Бессмертных Магов (по состоянию на 2004 год)

- Пинженина Софья Владимировна, Челябинск (Маг 1998/1999)
- Петров Владимир Анатольевич, Волгоград (Маг 1998/1999)
- Рудь Алексей Владиславович, Снежинск (Маг 1999/2000, 2000/2001)
- Семенова Наталья Евгеньевна, Тольятти (Маг 2001/2002)
- Есауленко Оксана Николаевна, Качканар (Маг 2001/2002)
- Вассина Роза Николаевна, Качканар (Маг 2001/2002)
- Пранис Сергей Александрович, Качканар (Маг 2001/2002)
- Семенко Виктор Петрович, Рубцовск (Маг 2002/2003)
- Шумилина Нина Дмитриевна, Тверь (Маг 2003/2004)
- Матыкин Вячеслав Юрьевич, Новохопёрск (Маг 2003/2004)

С 2003/2004 учебного года звание «Маг программирования» стало присуждаться и детям, которые показали особые успехи на роботландских сетевых турнирах.

Эти славные имена Магов от программирования навечно занесены в историю Роботландского университета.

- Мусин Марат Нариманович, Тверь, 9 класс, МОУ «Тверская гимназия № 6» (Маг 2003/2004)
- Громова Анна Валерьевна, Тверь, 9 класс, МОУ «Тверская гимназия № 6» (Маг 2003/2004)
- Погорелая Дарья Андреевна, Миасс, 6 класс, школа № 19 (Маг 2003/2004)
- Кротенко Владимир Николаевич, Алтайский край, г. Рубцовск, 8 класс, школа № 1 (Маг 2003/2004)

- **Югатов Владимир Анатольевич**, Алтайский край, г. Рубцовск, 8 класс, школа № 1 (Маг 2003/2004)
- **Носов Алексей Юрьевич**, Алтайский край, г. Рубцовск, 8 класс, школа № 1 (Маг 2003/2004)

## История Кукарачи

Исполнитель Кукарача (первое имя Таракан) родился на Дальнем Востоке в 1984 году. Роботландии тогда ещё не было, а был новосибирский Муравей, придуманный Г. А. Звенигородским вместе с группой исследователей (в неё входил и Ю. А. Первин, будущий директор Роботландии).

Второй участник будущей Роботландии, А. А. Дуванов в то время работал преподавателем в Благовещенском пединституте. Интерес благовещенцев к пионерским работам новосибирцев был велик. Настолько, что Дальний Восток приехал в Сибирь пообщаться с умными людьми, увидеть уникальную работу школьников на компьютерах своими глазами. По трагическому стечению обстоятельств как раз в это время от гриппа умер Г. А. Звенигородский. Ему было 32 года. Он даже не увидел издание своей книги, которая вышла уже после его смерти (Г. А. Звенигородский. Первые уроки программирования. Библиотечка Квант. Выпуск 41. Москва. Наука. 1985).

Разговор состоялся с Юрием Абрамовичем Первиным. Был показан школьный урок в младших классах за компьютерами «Агат». Была подробная содержательная беседа о новосибирском опыте, в частности, об исполнителе Муравей.

В Благовещенске закипела работа. В группу энтузиастов, которую возглавил А. А. Дуванов, вошли: О. Г. Какаулин, В. В. Немилостива, Ю. В. Прашкович, О. Д. Десятириков.

Появилась идея: взяв за основу среду Муравья, придумать такую её модификацию и такой учебный язык, которые, с одной стороны, были бы предельно просты и доступны самым маленьким, а с другой — позволили бы представить все основные управляющие структуры.

Результат первых опытов в этом направлении — исполнитель Тараканчик, работающий в командном режиме. Олег Какаулин спустя 18 лет реконструировал среду Тараканчика. Вы можете посмотреть, как выглядел Тараканчик на ЭВМ ИСКРА-1256 (и даже поработать с ним) на следующей ссылке:

**[www.botik.ru/~robot/history/chick.htm](http://www.botik.ru/~robot/history/chick.htm)**

Затем появился Таракан с возможностью программного управления. Позже, после поездки А. А. Дуванова на Кубу, Таракан получил второе имя — Кукарача.

Придумав исполнителя, мы даже и не подозревали обо всех его необычных возможностях. Первотолчком, закрутившим Кукарачу на новых оборотах, стали рекурсивные задачи Е. П. Лилитко, которые он придумал для своей дочки и любезно передал нам. Потом было много других задач, неожиданно

радостных находок. Особый успех Кукарача приобрёл на курсе 31 Роботландского университета ([www.botik.ru/~robot/ru](http://www.botik.ru/~robot/ru)).

Турнирные задания на этом курсе всегда неожиданны, а эмоциональный всплеск конкурсов имеет в университете повышенный накал. Каждый год куратор придумывает задачу для конкурса в надежде, что никто из ребят её не решит. Но такого ещё не случалось. Школьники дошли до того, что спокойно пишут трансляторы и даже решили для Кукарачи задачу о Ханойской башне (Первушин Данил, Снежинск)! И всё это притом, что в языке нет ни одной переменной!

## История Корректора

Корректор, как и Кукарача, переехал в Переславль-Залесский из Благовещенска (дальневосточный город, центр Амурской области, граница с Китаем). Этот исполнитель был придуман А. А. Дувановым двумя годами позже Кукарачи, в 1986 году. В этом же году была подготовлена рукопись книги «Введение в программирование. Корректор».

Однако эта рукопись так и не появилась в печати. На следующий год Дуванов переехал из Благовещенска в Переславль-Залесский и поступил на работу в Институт программных систем АН СССР в лабораторию школьной информатики, которую возглавил, приехав в Переславль из Новосибирска, Ю. А. Первин.

В Переславле закипела работа над новыми идеями. Рукопись сначала просто была положена в стол, а потом на время забыта.

Тем не менее, Корректор был одной из самых первых программ, реализованных на Ямахе и вошедших в самую раннюю версию курса «Роботландии». В дальнейшем, при переносе «Роботландии» сначала на УКНЦ, а затем на РС, Корректор остался на старой квартире. Мы не стали включать Корректор в новые версии «Роботландии», потому что его программирование выходило за рамки общеобразовательного курса информатики для младших школьников, носило факультативный характер и предназначалось для детей, которые хотели заниматься именно программированием. На пристройку к Роботландии кружка юных программистов не было сил и времени: нужно было разрабатывать и опробовать в школьных классах ставшие теперь традиционными, роботландские общеобразовательные темы.

Только в 1992 году, когда для издательства «Педагогика-Пресс» готовилась книга «Необычайные приключения Пети Кука в Роботландии» (авторы А. Дуванов, Ю. Первин), рукопись по Корректору была извлечена из картонной коробки и использована при подготовке четырёх глав приключений главного героя.

Роботландский университет дал новую жизнь Корректору. Рекурсивные опусы Кукарачи у Корректора получили логическое продолжение, более оправданное в его псевдотьюринговской среде, чем на клетчатом поле коллеги.

# Глава 1



## Олимпиада 1996/1997

### Турнир Кукарачи

— Здравствуй, Пух, — сказала она. — Какие новости?

— Грустные и ужасные, — сказал Пух, — потому что Иа-Иа, мой старый друг, потерял свой хвост, и он очень убивается о нём. Будь так добра, скажи мне, пожалуйста, как мне его найти?

— Ну, — сказала Сова, — обычная процедура в таких случаях нижеследующая...

*А. Милн*

#### 1. Деление на два (3 балла).

Автор: Е. П. Лилитко.

Кукарача расположен в верхнем левом углу поля. Где-то под ним в первом столбце в строке с чётным номером находится кубик с произвольным символом. Требуется разделить номер строки, в которой стоит кубик, на два и поставить Кукарачу в соответствующую строку. Поле считать бесконечным вниз. Пример начального и конечного состояний среды показан на рис. 1.1.

#### 2. Ишак (4 балла).

Автор: А. А. Дуванов.

На одном из перевёрнутых кубиков во втором столбце — буква Ш. Найдите её и составьте слово ИШАК. Букву Ф с поля надо убрать. Пример начального и конечного состояний среды показан на рис. 1.2.



Рис. 1.1



Рис. 1.2



Рис. 1.3

## 3. Треугольник (5 баллов).

Автор: Е. П. Лилитко.

Первый столбец поля пуст. На остальном пространстве расположены два кубика, стоящие в одном столбце, но не в соседних строчках. Кукарача находится в первой позиции строки, содержащей первый кубик. Поставить Кукарачу так, чтобы он являлся третьей вершиной равнобедренного прямоугольного треугольника (справа от первого кубика). Поле считать бесконечным вправо и вниз. Пример начального и конечного состояний среды показан на рис. 1.3.

## 4. Прямоугольник (6 баллов).

Автор: Е. П. Лилитко.

Первый, предпоследний и последний столбцы поля  $10 \times 10$  пусты. Пусты также и две нижние строки. На остальном пространстве расположены буквы А, В и С так, что они образуют прямоугольный треугольник с размерами катетов, большими единицы. Кукарача находится в первой позиции строки, содержащей букву А. Требуется поставить Кукарачу так, чтобы он являлся четвёртой вершиной прямоугольника. Независимо от расположения букв (при точном соблюдении приведённых условий), сообщение «Не могу» не должно возникать никогда. Пример начального и конечного состояний среды показан на рис. 1.4.



Рис. 1.4

## 5. Мышь (4 балла).

Автор: А. А. Дуванов.

На одном из перевёрнутых кубиков в пятой строке поля Кукарачи — буква Ь. Найдите её и составьте слово МЫШЬ. В первом столбце третьей строки

расположена буква Ф — её надо спихнуть с поля. Пример начального и конечного состояний среды показан на рис. 1.5.



Рис. 1.5

6. Новый исполнитель (6 баллов).

Автор: Е. П. Лилитко.

Построить нового исполнителя, «управляемого данными», поведение которого будет полностью зависеть от расположения букв на поле. Изначально Кукарача движется вниз до тех пор, пока не встретит одну из букв В, Н, Л, П или С. Встретив одну из этих букв, он продолжает движение соответственно вверх, вниз, влево, вправо или немедленно останавливается. На дальнейшем пути Кукараче может вновь повстречаться одна из приведённых букв, и тогда он вновь изменяет направление вышеописанным способом. Для нового исполнителя создайте программу (установите в нужные места буквы-команды и Кукарачу), заставляющую его ходить по расширяющейся спирали.

7. Значение цифры (3 балла).

Автор: А. А. Дуванов.

Под Кукарачей во второй строке расположена цифра (её значение больше двух). Определить, какая это цифра, и поставить её в строку с соответствующим номером. Пример начального и конечного состояний среды показан на рис. 1.6.





Рис. 1.6

8. Количество букв А на поле (9 баллов).

Автор: Е. П. Лилитко.

Кукарача расположен в нижнем левом углу поля. Первая строка и первый столбец поля пусты за исключением буквы Г, расположенной точно в левом верхнем углу поля. На остальном пространстве поля разбросано некоторое количество букв А, не более, чем по одной на строку. Таким образом, некоторые строки содержат одну букву А, а некоторые — ни одной. Требуется очистить поле от букв, попутно подсчитывая их количество. В конце работы Кукарача должен остановиться в строке, номер которой равен количеству сброшенных с поля букв А. То есть, если изначально на поле была одна буква А, Кукарача останавливается в первой строке, если две — во второй и т. д. Сообщение «Не могу» допускается только в том случае, если ни одной буквы А на поле не оказалось.

9. Экран (8 баллов).

Автор: А. А. Дуванов.

Во второй строке злоумышленник перемешал буквы в слове ЭКРАН. Восстановить испорченное слово. После исправления слово может быть расположено в любой строке или даже столбце. В последнем случае его нужно читать сверху вниз. Пример начального и конечного состояний среды показан на рис. 1.7.

10. Кукарачья арифметика.

Автор: А. А. Дуванов.

Введём новые обозначения для команд исполнителя (рис. 1.8).

Программа для исполнителя теперь задаётся числом, состоящим из цифр 1, 2, 3, 4, например, короткий поход Кукарачи на рис. 1.9 запишется в виде программы так: 1234.



Рис. 1.7

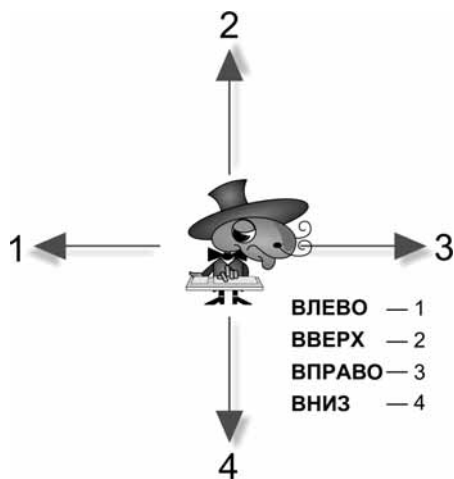


Рис. 1.8

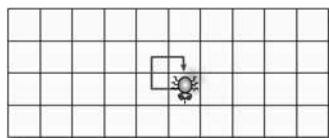


Рис. 1.9

Будем полагать также, что поле Кукарачи бесконечно (или, по крайней мере, очень большое).

- 1) (3 балла) Доказать, что если траектория движения приводит Кукарачу в клетку, с которой он начал движение, то сумма цифр числа, задающего программу исполнителя, — чётное число.
- 2) (2 балла) Верно ли обратное: если сумма цифр числа, задающего программу движения исполнителя — чётное число, то после выполнения программы Кукарача оказывается в исходной позиции.
- 3) (3 балла) Известно, что в программе исполнителя нет команд **ВВЕРХ**, а количество команд **ВЛЕВО** равно количеству команд **ВПРАВО**. Докажите, что в этом случае сумма цифр числа, изображающего программу, делится на 4, а если при этом условии ещё количество команд **ВЛЕВО** равно количеству команд **ВНИЗ**, то сумма цифр числа, изображающего программу, делится на 8.

## Глава 2



# Турниры 1997/1998

## 2.1. Турнир Кукарачи

— Вперёд! Вперёд! — кричал Кристофер Робин.  
— Вперёд! — кричали Пух и Пятачок.  
— Вперёд! — кричала Сова.  
— Тронулись! — сказал Кролик. — Я должен бежать. — И он помчался в голову колонны к Кристоферу Робину.  
— Вот именно, — сказал Иа. — Все тронулись. Но я тут ни при чём.

*А. Мшин*

### 1. Прогулка (3 балла).

Автор: А. А. Дуванов.

Помогите Кукараче совершить прогулку по полю размером  $10 \times 10$  из клетки (2,1) в клетку (1,10), не сдвигая кубиков. Начальное и конечное состояния среды показаны на рис. 2.1.

Дано		Надо	
	1 2 3 4 5 6 7 8 9 10		1 2 3 4 5 6 7 8 9 10
1	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
2	☀	Ж Ж	
3	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
4	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
5	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
6	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
7	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
8	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
9	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	Ж Ж Ж Ж Ж Ж Ж Ж Ж Ж	
10			

Рис. 2.1

## 2. Расставь цифры (4 балла).

Автор: А. А. Дуванов.

Исполнитель стоит в левом верхнем углу поля. Во второй строке, начиная с первой клетки, находится плотный ряд кубиков с цифрами. Цифры в ряду расположены в произвольном порядке и значение каждой из них больше двух. Поставить кубики в строки с номерами, равными значениям цифр. Считается, что поле исполнителя имеет 10 строк, а длина строки не ограничена. Пример начального и конечного состояний среды показан на рис. 2.2.



Рис. 2.2

## 3. Сложение двух палочных чисел (5 баллов).

Автор: А. А. Дуванов.

Будем обозначать целые положительные числа палочками (символ «|»): через | — обозначим число 1, через || — число 2, через ||| — число 3 и т. д. Иными словами, целое положительное число  $n$  будем записывать как группу из  $n$  палочек, идущих подряд в  $n$  соседних клетках одной строки. На основе введённых обозначений рассмотрим операцию сложения двух чисел. Понятно, что число палочек, обозначающих результат, равно сумме палочек, изображающих слагаемые.

А теперь заставим поработать Кукарачу, который в начальный момент находится в клетке (3,1). Во второй строке, начиная с первой клетки, записан пример на сложение двух палочных чисел (на рис. 2.3 изображён один из возможных вариантов). В конце работы программы на поле должен остаться результат вычислений. Поле исполнителя считается бесконечным вправо. Пример начального и конечного состояний среды показан на рис. 2.3.

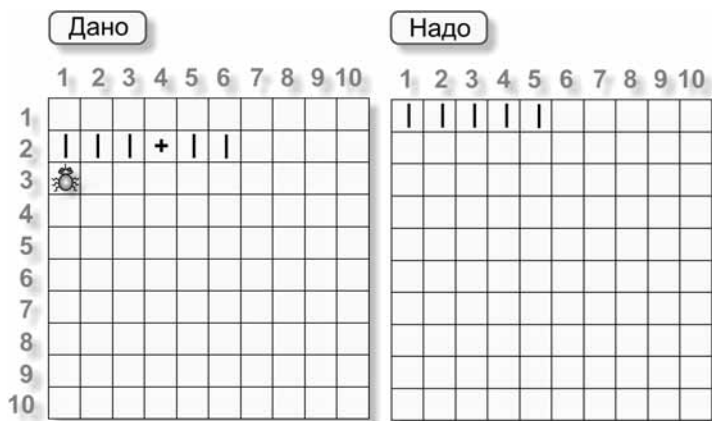


Рис. 2.3

## 4. Вычитание двух палочных чисел (10 баллов).

Автор: А. А. Дуванов.

В третьей строке, начиная со второй клетки, записан пример на вычитание двух палочных чисел. Кукарача находится в клетке (4,2). Вычислить результат, если известно, что вычитаемое не больше уменьшаемого. Поле исполнителя считается бесконечным вправо. Пример начального и конечного состояний среды показан на рис. 2.4.



Рис. 2.4

## 5. Деление палочного числа на 2 (5 баллов).

Автор: А. А. Дуванов.

В первой строке поля, начиная с первой позиции, записано чётное палочное число. Выполнить деление его на два. В начале работы программы