

**Александр Дуванов  
Алексей Рудь  
Виктор Семенко**

**АЗЫ  
ПРОГРАММИРОВАНИЯ  
ФАКУЛЬТАТИВНЫЙ КУРС**

**КНИГА ДЛ Я УЧЕНИКА**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06(075.3)  
ББК 32.973я721  
Д79

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

Д79 Азы программирования. Факультативный курс. Книга для ученика. — СПб.: БХВ-Петербург, 2005. — 352 с.: ил.

ISBN 5-94157-583-1

Теоретические основы программирования излагаются в наглядных средах исполнителей Кукарача и Корректор, сложные темы становятся простыми и ясными.

Особое внимание уделяется основам формализации и построения алгоритмов, тестированию и отладке программ. Целый раздел посвящен принципам построения трансляторов. Программные среды исполнителей помещены на диске, прилагаемом к книге учителя из этого комплекта, который также содержит решения всех задач и дополнительные материалы к урокам. Закрепить практический опыт программирования поможет книга «Азы программирования. Задачник».

*Для учащихся средних образовательных учреждений*

УДК 681.3.06(075.3)  
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>Иины 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. 28,38.

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 5-94157-583-1

© 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>11</b>
<b>Обращение к читателю.....</b>	<b>11</b>
Авторство задач и решений.....	13
Авторство иллюстраций.....	13
<b>Вступление .....</b>	<b>15</b>
Понятие исполнителя.....	15
Понятие алгоритма и программы.....	16
<b>ЧАСТЬ I. КУКАРАЧА.....</b>	<b>19</b>
<b>Глава 1. Кукарача и его среда обитания.....</b>	<b>21</b>
1.1. Знакомство с Кукарачей.....	21
Вопросы и задания .....	24
1.2. Первая задача.....	25
Задача .....	25
1.3. Задачи .....	26
<b>Глава 2. Вася экономит свой труд .....</b>	<b>29</b>
2.1. Ток — это кот задом наперёд.....	29
2.2. У компьютера есть память, и это хорошо .....	32
Вопросы и упражнения.....	34
2.3. Неприятное свойство компьютерной памяти, которое исправляется наличием дисков.....	35
2.4. Программа может содержать много процедур .....	35
2.5. Задачи .....	36

<b>Глава 3. Новые команды и их повторение</b> .....	<b>39</b>
3.1. Кукарача говорит «Ах!» .....	39
Вопросы и задания .....	41
3.2. Процедурное программирование .....	41
Задача «НАРЫЧАЛО» .....	42
Вопросы и задания .....	45
3.3. Команда повторения .....	45
Вопросы и задания .....	46
3.4. Интерпретатор и его странные сообщения .....	47
Вопросы и задания .....	50
Задача .....	51
Решение .....	51
3.5. Задачи .....	52
<b>Глава 4. Кукарача на распутье</b> .....	<b>55</b>
4.1. Команда ветвления .....	55
Задача 1 .....	56
Задача 2 .....	59
Вопросы и задания .....	60
4.2. Особые случаи .....	61
Задача 3 .....	61
Задача 4 .....	62
Задача 5 .....	62
Задача 6 .....	63
Вопросы и задания .....	65
4.3. Задачи .....	67
<b>Глава 5. Другой тип повторения</b> .....	<b>71</b>
5.1. Когда неизвестно число повторений .....	71
Задача 1 .....	73
Задача 2 .....	74
Вопросы и задания .....	76
5.2. Как обмануть интерпретатор .....	77
Задача 3 .....	78
Вопросы и задания .....	79
5.3. Задачи .....	79

<b>Глава 6. Кукарача хочет укусить себя за хвост .....</b>	<b>81</b>
6.1. Разговор о рекурсии .....	81
Задача 1 .....	82
6.2. Рекурсивный практикум .....	86
Задача 2 .....	86
6.3. Головоломное программирование .....	87
Задача 3 (автор Лилитко Е. П.) .....	87
6.4. Подводные камни .....	91
6.5. Задачи .....	93
<b>Глава 7. Задачи .....</b>	<b>97</b>
Задачи недели 2002/2003 учебного года .....	98
Задачи к главе 4 .....	98
Задачи к главе 5 .....	101
Задачи к главе 6 .....	103
Задачи недели 2003/2004 учебного года .....	104
Задачи к главам 1–3 .....	104
Задачи к главам 4–5 .....	106
Задачи к главе 6 .....	111
<b>Часть II. Корректор .....</b>	<b>113</b>
<b>Глава 8. Знакомство с исполнителем .....</b>	<b>115</b>
8.1. Корректор и его среда обитания .....	116
Вопросы и упражнения .....	119
8.2. Попробуем управлять .....	120
Вопросы и упражнения .....	121
8.3. Управление при помощи программы .....	121
Задача .....	122
Вопросы и упражнения .....	124
<b>Глава 9. Язык программирования .....</b>	<b>125</b>
9.1. Процедурное программирование .....	126
Задача 1 .....	126
Вопросы и упражнения .....	129
9.2. Циклы .....	130
Задача 2 .....	131
Задача 3 .....	132

Задача 4 .....	133
Задача 5 .....	134
Вопросы и упражнения .....	134
9.3. Развилки .....	136
Задача 6 .....	137
Задача 7 .....	138
Вопросы и упражнения .....	139
9.4. Рекурсия .....	141
Вопросы и упражнения .....	141
Справочник по языку программирования .....	142
Разделитель слов .....	142
Программа .....	142
Процедура .....	142
Имя процедуры .....	143
Комментарии .....	143
Команды .....	143
Справочник по условиям Корректора .....	144
<b>Глава 10. Отладка программ.....</b>	<b>145</b>
10.1. Корректор развлекается с котом .....	145
Задача 1 .....	146
10.2. Синтаксические ошибки .....	149
Вопросы и упражнения .....	151
10.3. Ошибки программирования .....	152
Вопросы и упражнения .....	155
10.4. Тестирование .....	156
10.5. Задачи .....	158
<b>Глава 11. Приёмы программирования Корректора .....</b>	<b>161</b>
11.1. Как найти конец текста .....	161
Задача 1 .....	162
Вопросы и упражнения .....	163
11.2. Как вернуться в исходное место .....	164
Задача 2 .....	164
Вопросы и упражнения .....	168
11.3. Специальные символы-флаги .....	168
Задача 3 .....	169
11.4. Задачи .....	175

<b>Глава 12. Арифметика чисел, палочек и символов .....</b>	<b>177</b>
12.1. Арифметика чисел .....	178
Задача 1 .....	178
Задача 2 .....	183
Задача 3 .....	186
Рекурсия или не рекурсия? .....	187
Задачи .....	188
12.2. Арифметика палочек .....	190
Задача 4 .....	190
Задача 5 .....	192
Задачи .....	196
12.3. Арифметика символов.....	197
Перевод символьного числа в обычное число .....	199
Алгоритм перевода символьного числа в обычное число .....	200
Перевод обычного числа в символьное число .....	201
Пример использования символьных чисел .....	202
Задача 6 .....	202
Задачи .....	203
<b>Глава 13. Преобразования, подсчёты, редактирование.....</b>	<b>207</b>
13.1. Длина текста.....	207
Задача 1 .....	208
Задача 2 .....	211
Задача 3 .....	212
Задачи .....	213
13.2. Корректор оправдывает своё имя .....	214
Задача 4 .....	214
Задача 5 .....	216
Задача 6 .....	220
Задачи .....	222
<b>Глава 14. Трансляторы.....</b>	<b>225</b>
14.1. Проверка объектов .....	225
Задача 1 .....	226
Задача 2 .....	228
Задачи .....	229
14.2. Транслятор для Плюсика.....	231
Исполнитель Плюсик.....	231
Задача 3 .....	235
Задачи .....	239

<b>Глава 15. Задачи</b> .....	<b>243</b>
Задачи недели 2002/2003 учебного года.....	243
Задачи, рекомендуемые к главам 11 и 12.....	243
Задачи, рекомендуемые к главам 12 и 13.....	244
Задачи недели 2003/2004 учебного года.....	246
Задачи к главам 8 и 9.....	246
Задачи к главе 10.....	249
Задачи к главе 11.....	251
Задачи к главе 12.....	253
 <b>Часть III. Транслятор?.. Это очень просто!</b> .....	 <b>257</b>
 <b>Глава 16. Язык Бэкуса-Наура</b> .....	 <b>259</b>
16.1. Понятие метаязыка.....	259
16.2. Определения на языке Бэкуса-Наура.....	259
Определение 1.....	259
Определение 2.....	260
Определение 3.....	260
16.3. Математическая индукция.....	261
16.4. Задачи.....	262
Определение 4.....	262
Определение 5.....	263
Определение 6.....	263
Определение 7.....	264
 <b>Глава 17. Кукарача и лексический анализ выражений</b> .....	 <b>265</b>
17.1. Транслятор.....	265
17.2. Лексема.....	266
17.3. Диаграмма переходов.....	266
Определение 1.....	267
17.4. Программа для Кукарачи.....	269
Постановка задачи.....	269
Программа.....	270
17.5. Задачи.....	272
Определение 2.....	272
Определение 3.....	273
Определение 4.....	274
Определение 5.....	275



<b>Глава 18. Ах уж эта рекурсия!</b> .....	<b>277</b>
18.1. Рекурсивная пружинка .....	278
Определение 1 .....	278
18.2. Задачи.....	281
Определение 1 .....	281
Определение 2 .....	282
Определение 3 .....	283
<b>Глава 19. Лексический анализатор в среде Корректора</b> .....	<b>285</b>
19.1. Постановка задачи.....	286
Определение 1 .....	286
Решение .....	287
19.2. Задачи.....	289
Определение 2 .....	289
Определение 3 .....	290
Определение 4 .....	291
<b>Глава 20. Построение трансляторов</b> .....	<b>293</b>
20.1. Интерпретаторы и компиляторы.....	293
Компилятор .....	293
Интерпретатор.....	293
20.2. План работы .....	293
20.3. Простой транслятор .....	294
Задача 1 .....	294
Решение .....	295
Задача 2 .....	299
20.4. Построение компилятора .....	300
Задача 3 .....	300
Запись выражения по-польски и стековые вычисления .....	303
Программа для Корректора.....	312
<b>Ссылки на задачи</b> .....	<b>323</b>
Часть I. Кукарача.....	323
Часть II. Корректор.....	331
Часть III. Транслятор?.. Это очень просто .....	339
<b>Предметный указатель</b> .....	<b>341</b>

## 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).

## Обращение к читателю

— А нам и так хорошо живётся! — кричит нервный Шарик.  
— Вам плохо живётся, — объясняет тётя. — Только вы этого не понимаете. Вы по ошибке счастливы. Но я вам глаза раскрою.

*Э. Успенский*

Эта книга — отражение самого творческого курса Роботландского сетевого университета, на котором школьники 5–8 классов приступают к изучению основ программирования. Ребята быстро забывают про то, что они ученики, становятся вровень со своими наставниками, а порой заставляют учителей испытывать зависть найденными решениями и новыми потрясающими задачами!

Программирование — удивительный род человеческой деятельности, который сродни волшебству. Несколько заклинаний на языке посвящённых — и «твёрдый» металл на письменном столе получает «мягкую» душу: компьютер оживает и ведёт себя «разумно». Переживания программиста, вероятно, сравнимы с переживанием Создателя. Ведь Он сотворил живых тварей при

помощи одного Своего слова. Наверное, фраза «и создал Он их по образу и подобию Своему» относится как раз к программистам. Если вы понимаете, о чём я говорю, значит, в душе вы — программист и взяли в руки эту книгу не зря!

### **О волшебстве**

Первый раз автор столкнулся с волшебством в 6 классе. Брат подарил мне книгу Е. Айсберга «Радио?.. Это очень просто!». Свой первый радиоприёмник я спаял на деревянной дощечке, которую выпилил ножовкой из доски, валявшейся в сарае. Пригладил её рубанком, отполировал шкуркой, приклеил клеем БФ транзисторы, конденсаторы, резисторы... Потом спаял всё это согласно схеме. Какой же была моя радость пополам с удивлением, когда эта деревяшка заговорила человеческим голосом! Своими руками я сотворил Буратино!

Стало понятно, что я волшебник. Это было удивительное ощущение! Второй раз испытал подобные чувства, когда написал первую программу. А потом каждый раз, создавая программные коды, не переставал удивляться (не перестаю и сейчас!), как можно словесными заклинаниями вдохнуть жизнь в железного (теперь) Буратино на моём рабочем столе!

Мне хочется передать эту радость творчества и это постоянное удивление нынешним ребятам и девочкам! Это желание — основная причина появления этой книги.

Программисты — счастливые люди, но жизнь у них трудна. Во-первых, нужно знать теорию волшебства, во-вторых — язык, на котором записываются заклинания, и, в-третьих — знать исполнителя, который эти заклинания выполняет, превращая их в ощущение жизни на экране.

В данной книге программирование — это:

1. Теория — основы построения алгоритмов.
2. Исполнитель — Кукарача, Корректор.
3. Язык — язык роботландских исполнителей.

Составляющие знаний программиста пронумерованы в порядке убывания важности.

Самым неважным оказывается язык, хотя новичок всегда думает, что язык — это и есть программирование. Это всё равно, как сказать, что писательство — это авторучка, а химия — это пробирки. Или что топор — это основа плотницкого мастерства. Конечно, для писателя важно знать, как менять стержень в авторучке, а для плотника — как затачивать топор. Но знания топора и авторучки не лежат в основе изделий, которые вырубаются из дерева и выписываются на бумаге.

Так и в программировании: язык — это второстепенный элемент, при помощи которого описываются наши знания основ построения алгоритмов в виде программы для конкретного исполнителя.

Если знаний основ формализации и построения алгоритмов нет, изучение языка программирования их не прибавит.

Как говорил классик программирования Эдгар Дийкстра — самый главный язык, который должен знать программист, — свой родной, на котором он изъясняется в повседневной жизни.

Иными словами, суть программирования не в языке программирования, а в умении чётко сформулировать задачу, выдвинуть идею решения, разработать алгоритм! И только потом можно перевести алгоритм в программу, записав несколько заклинаний на языке посвящённых!

Желаю приятного путешествия по страницам этой книги!

Для работы вам понадобится программная реализация исполнителей, описанных в книге. Этот продукт записан на диске, который прилагается к учительской части «Азов программирования».

Под отдельную обложку вынесен сборник задач, сформированный по материалам роботландских турниров. Если вы решились на покупку этой книги, то и задачник вам совершенно необходим. Ведь программирования без практики не бывает!

Автор исполнителей  
Кукарача и Корректор,  
руководитель Роботландского университета  
Александр Александрович Дуванов  
[kurs@robotland.pereslavl.ru](mailto:kurs@robotland.pereslavl.ru)

## Авторство задач и решений

Условие каждой задачи и каждое решение сопровождаются в книге ссылкой на автора этой задачи и этого решения. Если такой ссылки нет, то это означает, что автором задачи и (или) решения является А. А. Дуванов.

## Авторство иллюстраций

Автором всех иллюстраций является А. А. Дуванов. Для построения рисунков были использованы художественные заготовки А. А. Русса (художник Роботландии).

# Вступление

Было когда-то на свете двадцать пять оловянных солдатиков. Все они были сыновьями одной матери — старой оловянной ложки — и, значит, друг другу родными братьями. Они были очень красивы: ружьё на плече, грудь колесом, мундир красный с синим. Чудо что за солдатика!

Они лежали, все двадцать пять, в картонной коробке. В ней было темно и тесно. Но вот однажды коробка открылась.

*Г.-Х. Андерсен*

Ниже приводятся краткие объяснения понятиям *исполнитель*, *СКИ*, *среда*, *отказ*, *алгоритм*, *программа*, *транслятор*.

Эти базовые понятия будут уточняться и конкретизироваться на примерах работы с исполнителями Кукарача и Корректор. Сейчас просто послушайте, как звучат главные слова программистской магии!

## Понятие исполнителя

**Исполнителем** называют устройство, которое умеет выполнять команды из фиксированного списка.

Этот набор «понятных» исполнителю команд называют системой команд этого исполнителя, или сокращенно **СКИ**.

Обстановку, в которой работает исполнитель, называют его **средой**.

Различные исполнители получают свои команды по-разному. Управляя автомобилем, например, надо нажимать на педали, дергать за рычаги и крутить руль. А исполнитель-курсор на экране компьютера получает свои команды **влево**, **вправо**, **вверх**, **вниз**, когда человек нажимает на клавиатуре клавиши со стрелками или перемещает по коврику манипулятор мышь.

Управление исполнителем может привести к одной из двух **аварийных ситуаций (отказов)**, одна из них условно называется «**Не понимаю**», другая — «**Не могу**».

Отказ «Не понимаю» возникает у всех исполнителей в одном случае — задаётся команда, не входящая в СКИ.

Отказ «Не могу» специфичен для каждой команды из СКИ — он возникает, когда условия среды не позволяют выполнить заданную команду (автомобиль не может ускориться от педали газа, если не включена передача, курсор не может выйти за пределы экрана).

Для знакомства с исполнителем можно использовать схему, изображённую на рис. 1.

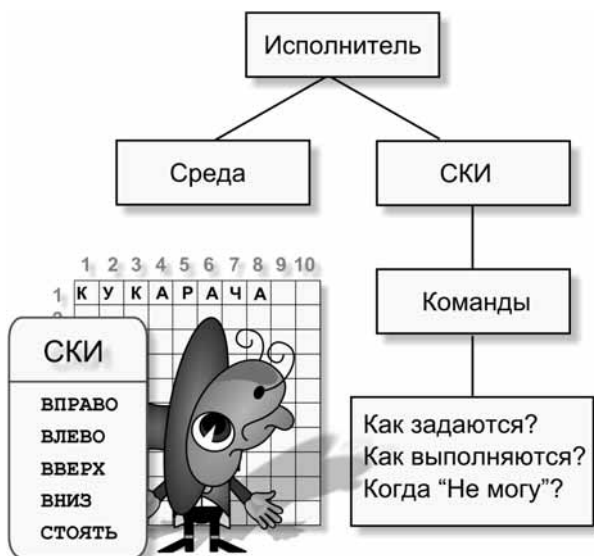


Рис. 1. Схема знакомства с исполнителем

## Понятие алгоритма и программы

**Алгоритм** — это описание действий исполнителя для выполнения определённой работы.

Чтобы алгоритм стал понятен исполнителю, его записывают при помощи команд из СКИ. Такая запись алгоритма называется **программой**.

Для записи программ часто пользуются специальными **языками программирования**. В эти языки наряду с командами СКИ входят **процедуры**, описывающие новые команды (команды, которые конструируются программистом), и **управляющие структуры**, предназначенные для записи алгоритма.

Исполнитель понимает команды только из своей СКИ и выполнить программу, написанную на языке программирования, самостоятельно не может.

Для «перевода» текста программы в СКИ строят новый исполнитель, называемый **транслятором** (**интерпретатор** или **компилятор** языка). Процесс выполнения программы, написанной на языке программирования, упрощённо поясняет схема, изображённая на рис. 2.

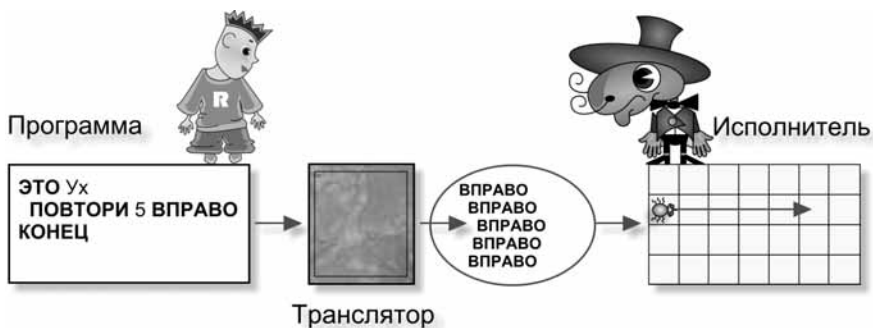


Рис. 2. Схема выполнения программы



## **Часть I**

# **Кукарача**

**Глава 1. Кукарача и его среда обитания**

**Глава 2. Вася экономит свой труд**

**Глава 3. Новые команды  
и их повторение**

**Глава 4. Кукарача на распутье**

**Глава 5. Другой тип повторения**

**Глава 6. Кукарача хочет укусить себя  
за хвост**

**Глава 7. Задачи**



# Глава 1



## Кукарача и его среда обитания

### 1.1. Знакомство с Кукарачей

— Как тебя зовут?

Кот говорит:

— И не знаю как. И Барсиком меня звали, и Пушком, и Оболтусом. И даже Кис Кисычем я был. Только мне всё это не нравится. Я хочу фамилию иметь.

*Э. Успенский*

**Вася.** Послушай, Петя, что это за приложение со странным названием «Кукарача» «живёт» на моём компьютере (рис. 1.1)?



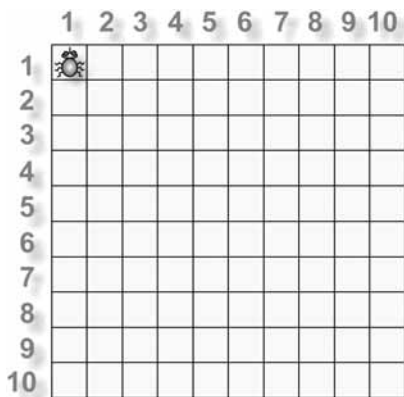
**Рис. 1.1.** Разговор братьев

**Петя.** Кукарача? Это исполнитель, который может ползать по клетчатому полю и собирать из кубиков разные слова. Давай я покажу тебе, как с ним работать.

**Вася.** Отлично! Запускаю Кукарачу.

**Петя.** Что ты видишь на экране?

**Вася.** Я вижу пустое клетчатое поле и на нём одинокое существо в левом верхнем углу (рис. 1.2).



**Рис. 1.2.** Клетчатое поле с исполнителем

**Петя.** Это и есть среда исполнителя. А в углу в ожидании указаний стоит сам исполнитель — Кукарача. На рис. 1.3 показана его СКИ (Система Команд Исполнителя).



**Рис. 1.3.** Система команд Кукарачи

Давай расскажу, как исполнитель выполняет команды. По командам **влево**, **вправо**, **вверх**, **вниз** он переползает в соседнюю клетку в указанном направлении. Команда **стоять** — это «пустая» команда. Выполняя её, Кукарача не делает никаких движений, он вообще ничего не делает!

**Вася.** Зачем же было включать такую команду в СКИ?

**Петя.** Она удобна при программировании развилок. Мы этим займёмся чуть позже.

**Вася.** А как задавать команды исполнителю?

**Петя.** Надо их записывать в редакторе команд и нажимать экранную кнопку *GO* (*go* по-английски означает «иди», читается «гоу») (рис. 1.4).



**Рис. 1.4.** Редактор команд

**Вася.** Хорошо, напишу команду **ВПРАВО** и нажму кнопку *GO*... Есть! Кукарача послушно сместился на одну клетку вправо (рис. 1.5).



**Рис. 1.5.** Выполнена команда **ВПРАВО**

А теперь скомандую **ВВЕРХ**. Ага, хитрый исполнитель пошевелил лапками и написал «Не могу» (рис. 1.6)!



**Рис. 1.6.** Исполнитель сообщает «Не могу»

**Петя.** Кукарача может сталкивать с поля кубики, но сам покинуть свою среду не может. Вот и сообщает об этом.

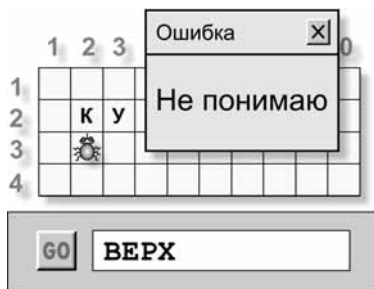
**Вася.** Но на поле нет никаких кубиков!

**Петя.** Их можно поставить. Кубик — это клетка, в которую записана буква, цифра или другой символ. Из кубиков можно складывать слова (рис. 1.7).

**Вася.** Я дал команду толкнуть вверх кубик **к**, а исполнитель не понял (рис. 1.8)!



**Рис. 1.7.** Кубики на поле исполнителя



**Рис. 1.8.** Исполнитель сообщает «Не понимаю»

**Петя.** Ты записал слово с ошибкой. Команды **ВЕРХ** нет в СКИ, и Кукарача не знает, что надо делать!

## Вопросы и задания

1. Что сделает Кукарача, получив команду **НАЛЕВО**?
2. Кукарача стоит в левом нижнем углу поля. Как он выполнит команду **ВНИЗ**?
3. Кукарача находится в клетке (1,1). Где он окажется после выполнения следующих команд?

- |           |           |           |           |
|-----------|-----------|-----------|-----------|
| а) ВПРАВО | б) ВПРАВО | в) ВПРАВО | г) ВПРАВО |
| ВНИЗ      | ВПРАВО    | ВНИЗ      | ВНИЗ      |
| ВПРАВО    | ВНИЗ      | ВЛЕВО     | ВЛЕВО     |
| ВНИЗ      | ВЛЕВО     | ВНИЗ      | СТОЯТЬ    |
| ВНИЗ      | ВЛЕВО     | ВНИЗ      |           |

## 1.2. Первая задача

### Приказ-расписание

1. Подъём в 7.30 (ответственный Матроскин).
2. Завтрак в 8.25 (ответственный Шарик).  
Поедание совместное.
3. Топка печки в 9.00 (ответственный Матроскин).
4. Доставка дров (ответственный Шарик).
5. Обед в 14.00 (ответственный Матроскин). Поедание совместное.
6. Мытьё посуды, но не облизывание, в 14.30 (ответственный Шарик).

*Э. Успенский*

**Петя.** Не пора ли перейти от общих слов к решению задач?

### Задача

Пусть Кукарача превратит слово КУЛИК в нашу фамилию (рис. 1.9).



**Рис. 1.9**

**Вася.** Ну, раз он может выталкивать с поля буквы, то это просто. Вот алгоритм решения этой задачи:

1. Выбросить лишние буквы.
2. Подойти к концу слова.
3. Придвинуть последнюю букву к началу слова.

**Петя.** Молодец! Первый пункт алгоритма предлагаю записать подробнее.

**Вася.** Можно. Тогда получится так:

1. Выбросить лишние буквы.
  - 1.1. Выбросить Л.
  - 1.2. Подойти к И.
  - 1.3. Выбросить И.

2. Подойти к концу слова.
3. Придвинуть последнюю букву к началу слова.

**Петя.** Начинай выполнять свой план.

Вася с увлечением приступил к управлению исполнителем, а Петя по давней привычке записал его работу в тетрадь (табл. 1.1):

**Таблица 1.1**

Шаги алгоритма	Команды	Среда исполнителя
1. Выбросить лишние буквы		
1.1. Выбросить Л	ВВЕРХ ВНИЗ	
1.2. Подойти к И	ВПРАВО	
1.3. Выбросить И	ВВЕРХ ВНИЗ	
2. Подойти к концу слова	ВПРАВО ВПРАВО ВВЕРХ	
3. Придвинуть последнюю букву к началу слова	ВЛЕВО ВЛЕВО	

## 1.3. Задачи

1. Превратить молоток в моток (рис. 1.10).



**Рис. 1.10**

2. Превратить мишку в мышку (рис. 1.11).



Рис. 1.11

3. Починить колесо (рис. 1.12).



Рис. 1.12

4. Какую роль хочет играть Кукарача на поле (рис. 1.13)?



Рис. 1.13

5. Получить имя знакомой кошки Кукарачи (рис. 1.14).



Рис. 1.14

## Глава 2



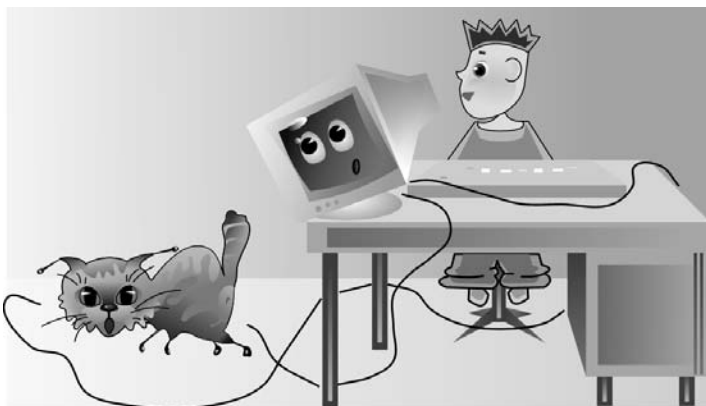
# Вася экономит свой труд

## 2.1. Ток — это кот задом наперёд

— Что-то у нас кошачьим духом пахнет.  
Не иначе как дядя Фёдор кота притащил.

*Э. Успенский*

**Петя.** Знаешь ли, Вася, что такое кот задом наперёд (рис. 2.1)?



**Рис. 2.1.** Что такое кот задом наперёд?

**Вася.** Кот — он и в огороде кот. Не понял вопроса!

**Петя.** А будет кот задом наперёд совсем не котом, а тем, что заставляет работать наш компьютер.

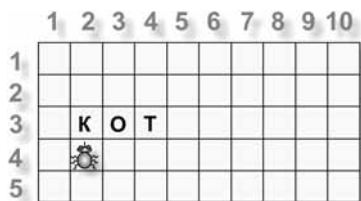
**Вася.** Процессором что ли?

**Петя.** Чудак, если слово кот прочитать с конца, то получится слово...

**Вася.** А, понял! Получится слово «ток».



**Петя.** Так вот, давай заставим Кукарачу превратить kota в ток (рис. 2.2).



**Рис. 2.2.** Превратить kota в ток

**Вася.** Надо переставить первую и последнюю буквы. Записываю алгоритм и команды (табл. 2.1):

1. Букву К во вторую строку.
2. Букву К в клетку (2,4).
3. Букву К на место, а букву Т в четвёртую строку.
4. Букву Т в клетку (4,2).
5. Букву Т на место.

Теперь записываю команды.

**Таблица 2.1**

Шаги алгоритма	Команды	Среда исполнителя
1. Букву К во вторую строку	ВВЕРХ	
2. Букву К в клетку (2,4)	ВЛЕВО ВВЕРХ ВПРАВО ВПРАВО	

Таблица 2.1 (окончание)

Шаги алгоритма	Команды	Среда исполнителя
3. Букву К на место, а букву Т в четвёртую строку	ВВЕРХ ВПРАВО ВНИЗ	
4. Букву Т в клетку (4,2)	ВПРАВО ВНИЗ ВНИЗ ВЛЕВО ВЛЕВО	
5. Букву Т на место	ВНИЗ ВЛЕВО ВВЕРХ	

**Петя.** Прекрасно! Кукарача вернулся на то самое место, с которого начал свою работу. Это значит, что если мы сейчас повторим все команды с самого начала и до конца, то...

**Вася.** Ток опять превратится в кота, но только я бы поленился снова набирать все 16 команд ради удовольствия вернуть хвостатого на место. В конце концов, перед нами компьютер, у которого, как ты сам мне говорил, есть память. Почему мы её не используем, а вместо этого эксплуатируем мой детский труд?

## Вопросы и задания

1. Изменится ли состояние среды Кукарачи, если Петя и Вася захотят переставлять местами буквы Н и С в слове СОН? Изменится ли программа? Почему? Заставьте Кукарачу сделать требуемую перестановку.
2. Потребуется ли новая программа, если ребята захотят переставить первую и последнюю буквы в слове СОРТ? Почему?

## 2.2. У компьютера есть память, и это хорошо

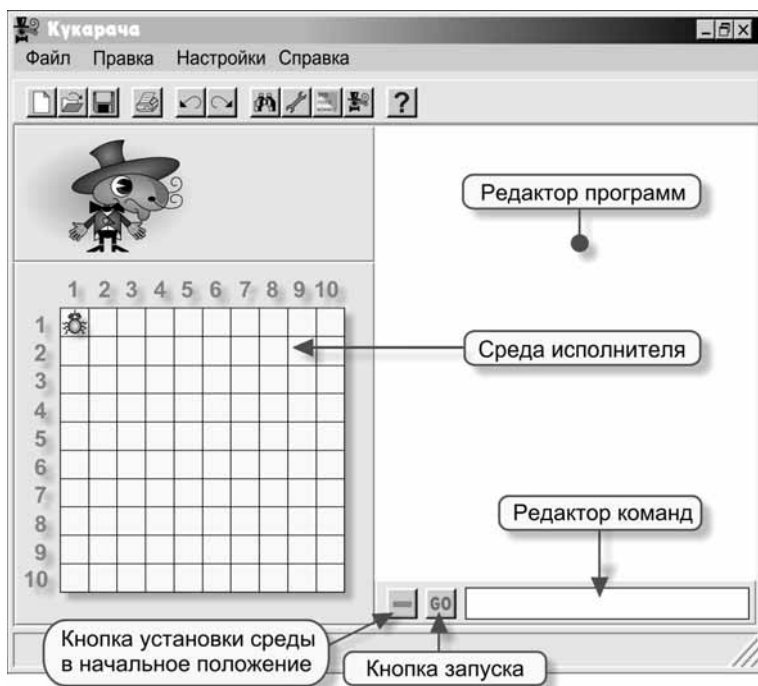
— На сегодня у нас такая программа намечается, — отвечает тётя. — Матроскина с Шариком мы бросаем в речку рыбу ловить. У нас по понедельникам будут рыбные дни. Пусть берут удочки и уходят удить.

*Э. Успенский*

На идею эксплуатации детского труда Петя отреагировал так.

**Петя.** Ну, ну, Вася, не заводись. Я всегда говорил, что лень — это двигатель прогресса. Что касается компьютерной памяти, тут ты прав. Сейчас покажу, как можно её использовать, решая алгоритмические задачи.

Вот справа на экране большое поле (рис. 2.3). Это редактор программ. В него можно один раз записать программу, а затем запускать её на выполнение столько раз, сколько потребуется, не набирая заново.



**Рис. 2.3.** Поле для записи программ

**Вася.** Хорошо, я наберу в этом поле программу, но как это сделать?

**Петя.** Войти в редактор программ можно, щёлкнув по его изображению мышкой, а программа записывается так:

**ЭТО** Имя

...

команды

...

**КОНЕЦ**

Слова **ЭТО** и **КОНЕЦ** называют **ключевыми** словами. Между ними записываются команды.

**Вася.** А что за имя нужно написать после слова **это**?

**Петя.** Имя твоей программы. Это имя становится названием новой команды для Кукарачи, хотя такая команда, понятно, не входит в СКИ. Написав программу, ты можешь запустить её на выполнение, задав имя в редакторе команд.

**Вася.** Получается, что я учу Кукарачу новым командам! А какое имя ты посоветуешь мне сейчас написать?

**Петя.** Имя может быть любым, только оно не должно совпадать с названиями команд из СКИ и ключевыми словами. Внутри имени нельзя писать пробелы.

Назови программу *Перевертыш*, ведь исполнитель переворачивает слово задом наперёд.

Вася согласился с братом и набрал в редакторе программ свой текст.

Петя попросил Васю писать **комментарии** при записи программы. Он объяснил, что комментарии не обрабатываются компьютером, они записываются для удобства программиста. С комментариями программа становится более понятной — человеку её легче читать и отлаживать.

Для записи комментариев в программах Кукарачи используют две косых черты «//». Они сами и всё, что находится правее до конца строки, не принимаются во внимание при выполнении программы (рис. 2.4).

**Петя.** Ну вот, программа записана. Теперь набери в редакторе команд её имя и нажми кнопку *GO*.

**Вася.** Записываю *Перевертыш*, нажимаю *GO* (рис. 2.5).

Да, здорово! Кукарача как заводная игрушка трудится на поле, и, что удивительно, сначала вырабатывает ток, а при повторном запуске возвращает кота на место!

**Петя.** Так получается, потому что перед повторным запуском среда не приведена в начальное положение. Если сейчас снова нажать кнопку *GO* (не нажимая кнопку возврата в начальное положение), то на поле опять получится слово «ток».

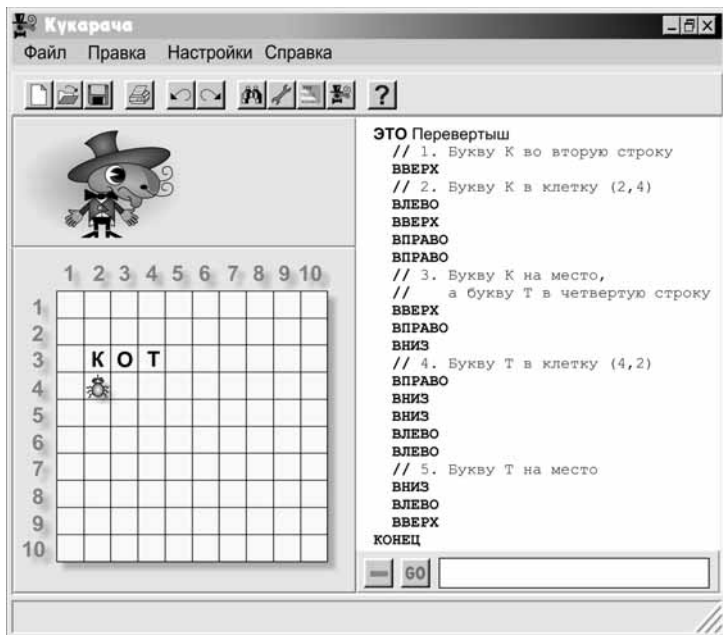


Рис. 2.4. Программа, записанная в программном поле

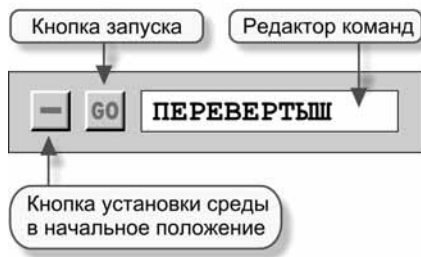


Рис. 2.5. Задана команда на выполнение

Вася щёлкает мышкой по кнопке *GO* и наблюдает за работой исполнителя.  
**Вася.** Ток, конечно, получился, но что будет, если я его выключу?

## Вопросы и упражнения

1. Что обозначают слова **это** и **конец** в программах Кукарачи?
2. Найдите ошибки в программе:

**Это** Очень хорошая команда

**СТОЯТЬ** (Кукарача стоит на месте)

**Конец.**

## 2.3. Неприятное свойство компьютерной памяти, которое исправляется наличием дисков

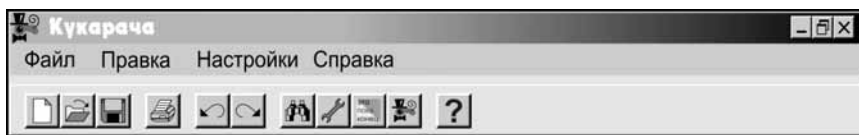
— Может быть, синяк ещё и не вскочит, — ответил Незнайка, смущённо вертя линейку в руках.

*Н. Носов*

**Петя.** Если ты выключишь компьютер, твой труд пропадёт, программа исчезнет из памяти, и завтра тебе придётся набирать её заново.

**Вася.** Помнится, ты намекал, что у компьютера есть магнитный диск для долговременного хранения информации. Говорил также, что информация хранится на диске подобно звуку на магнитной ленте нашего магнитофона. А ведь когда выключают магнитофон, на ленте ничего не пропадает.

**Петя.** Да, конечно, твою программу можно сохранить на диске, ты прав. Смыслённый у меня растёт брат! Меню окна практически любого компьютерного приложения содержит позицию *Файл* (рис. 2.6). Нужно щёлкнуть в этой позиции и сохранить работу на диске.



**Рис. 2.6.** Меню окна приложения «Кукарача»

**Вася.** Я запишу свою программу на диск под именем *kuk1*. С одной стороны, Кук — это моя фамилия, а с другой — сокращение от слова «Кукарача». Цифра «1» будет обозначать первую мою программу.

## 2.4. Программа может содержать много процедур

— Ну-ка, тащите сюда гаечный ключ, плоскогубцы, молоток и паяльник, — сказал Винтик малышам.

*Н. Носов*

**Вася.** Мне понравилось писать программы. С их помощью можно научить Кукарачу новым командам. Плохо только, что программу для каждой такой команды придётся записывать в отдельный файл.