

**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^{1/16}. 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

Состав комплекта	7
Вступление	9
Исполнители и язык программирования	9
Исполнитель Кукарача.....	9
Исполнитель Корректор	11
Организация обучения на курсе «Азы программирования»	15
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
Глава 1. Олимпиада 1996/1997.....	23
Турнир Кукарачи.....	23

Глава 2. Турниры 1997/1998	31
2.1. Турнир Кукарачи	31
2.2. Турнир Корректора	38
 Глава 3. Турниры 1998/1999	 45
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
 Глава 4. Турниры 1999/2000	 67
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
 Глава 5. Турниры 2000/2001	 85
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

Глава 6. Турниры 2001/2002	99
6.1. Турнир Кукарачи	99
Вступительное слово куратора	99
6.2. Турнир Корректора	106
Вступительное слово куратора	106
Глава 7. Турниры 2002/2003	113
7.1. Турнир Кукарачи	113
7.2. Турнир Корректора	117
Вступительное слово куратора	117
Глава 8. Турниры 2003/2004	125
8.1. Турнир Кукарачи	125
8.2. Турнир Корректора	131
Определения	131
Глава 9. Турниры 2004/2005	139
Олимпиада Кукарачи	139
Ссылки на задачи.....	145
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.

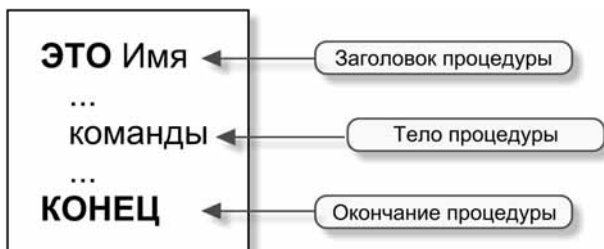


Рис. 5. Описание процедуры

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>	Команда исполнителя Выполнение определяется описанием команды в СКИ
<имя процедуры>	Вызов процедуры Выполняется процедура с указанным именем
ПОВТОРИ <число> <команда>	Цикл ПОВТОРИ Повторение выполнения команды указанное число раз
ПОКА <условие> <команда>	Цикл ПОКА Повторение выполнения команды, пока условие имеет значение <i>истина</i> . Проверка условия — перед выполнением команды
ЕСЛИ <условие> ТО <команда1> ИНАЧЕ <команда2>	Команда ветвления 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> }	Составная команда 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).

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.