

ОСНОВЫ алгоритмизации

Методические рекомендации

2010

УДК 53(075.3)+52(075.3)
ББК 22.3я72+22.6я72
К89

Авторы: преподаватель кафедры экономики и менеджмента УО ФПБ ВФ «МИТСО»
Д.Р. Кузьмичев; заведующий кафедрой информатики и ИТ УО «ВГУ им. П.М. Машерова», кандидат физико-математических наук, доцент **А.И. Бочкин**

Рецензент:
заведующий кафедрой геометрии и математического анализа УО «ВГУ им. П.М. Машерова»,
кандидат физико-математических наук, доцент *С.А. Шлапаков*

Методические рекомендации по основам алгоритмизации дополняются аннотированным списком литературы, предназначенной для подготовки студентов к педагогической работе.

Учебное издание адресуется студентам-выпускникам математического факультета, учителям школ, лицеев, гимназий, профессионально-технических учебных заведений.

УДК 53(075.3)+52(075.3)
ББК 22.3я72+22.6я72

© Кузьмичев Д.Р., Бочкин А.И., 2010
© УО «ВГУ им. П.М. Машерова», 2010

ОГЛАВЛЕНИЕ

Введение.....	4
Глава 1. Исполнители и алгоритмы	5
1.1. Историческая справка	5
1.2. Понятие «алгоритма»	5
1.3. Свойства алгоритмов	6
1.4. Исполнители для школьников.....	7
Глава 2. Формирование алгоритмического мышления.....	9
2.1. Комплексы задач разного уровня сложности для формирования понятия «событие» и алгоритмического стиля мышления школьников	9
Перевозчик.....	9
Переправы без условий (событий).	9
Переправляющиеся находятся на одном берегу.	9
Переправы без условий.....	10
Переправляющиеся находятся на разных берегах.....	10
Двухместная лодка (есть событие – лодка наполнена)	11
Трехместная лодка	12
Переправы с условиями. Условия на вместимость лодки.....	12
Переливашка.....	13
Задачи на деление некоторого количества жидкости с помощью двух дополнительных пустых сосудов за наименьшее число переливаний.	13
Задачи на получение некоторого количества жидкости из большого или бесконечного по объему сосуда, водоема или источника с помощью двух пустых сосудов	15
Глава 3. Исполнитель, управляемый событиями	16
3.1. Основные результаты. Функциональные возможности исполнителя	16
Глава 4. Программная реализация исполнителя	19
4.1. Основные глобальные переменные	19
4.2. Процедура <code>Make_field()</code>	19
4.3. Процедура <code>Exec_ev()</code>	20
4.4. Процедура <code>move()</code>	21
4.5. Выполнение кода.....	21
Приложение А. План-конспект урока	23
1. Содержательно-деятельностный компонент (ход урока).....	24
Словарь терминов	28
Дидактические материалы	28
Приложение Б. Дополнительные задачи к уроку	29
Задача 1.	29
Задача 2. «Задача Льва Толстого про шапку»	29
Задача 3. «16 команд»	30
Задача 4. «Что произойдет в городке?»	30
Задача 5. «Загадка Эйнштейна».....	30
Задача 6. Три ящика	31
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	32

Введение

Данное учебное издание предназначено для проведения практических, лабораторных, самостоятельных занятий студентов математического факультета.

Представленный материал разбит на части: в первом разделе вводится понятие алгоритма и исполнителя; во второй части представлен общий круг задач на развитие алгоритмического мышления, рассмотрен учебный комплекс для конкретных исполнителей – Робота и Чертежника [14] и Робота*, управляемого событиями [2]. Далее идет подробный анализ исполнителя Робот* и рассматриваются основные методические моменты в разработке алгоритма. В приложении представлен план-конспект урока по теме «Алгоритмизация».

Глава 1. Исполнители и алгоритмы

1.1. Историческая справка

Слово «алгоритм» происходит от имени великого среднеазиатского ученого VIII–IX вв. Аль-Хорезми (Хорезм – историческая область на территории современного Узбекистана). Из математических работ Аль-Хорезми до нас дошли только две – алгебраическая (от названия этой книги родилось слово алгебра) и арифметическая. Вторая книга долгое время считалась потерянной, но в 1857 г. в библиотеке Кембриджского университета был найден ее перевод на латинский язык. В ней описаны четыре правила арифметических действий, практически те же, что используются и сейчас. Так имя Аль-Хорезми перешло в Алгоритми, откуда и появилось слово «алгоритм». Термин «алгоритм» употреблялся для обозначения четырех арифметических операций, именно в таком значении он и вошел в некоторые европейские языки. Например, в авторитетном словаре английского языка Webster's New World Dictionary, изданном в 1957 г., слово «алгоритм» снабжено пометкой «устаревшее» и объясняется как выполнение арифметических действий с помощью арабских цифр.

«Алгоритм» вновь стал общеупотребительным с появлением электронных вычислительных машин для *обозначения совокупности действий, составляющих некоторый процесс*. Здесь подразумевается не только процесс решения некоторой математической задачи, но и любая инструкция, и многие другие последовательные правила, не имеющие отношения к математике.

На протяжении многих веков понятие «алгоритма» связывалось с числами и относительно простыми действиями над ними, да и сама математика была по большей части наукой о вычислениях. Чаще всего алгоритмы представлялись в виде математических формул.

1.2. Понятие «алгоритма»

В повседневной жизни каждый человек сталкивается с необходимостью решения задач самой разной сложности. Некоторые из них трудны и требуют длительных размышлений для поиска решений (а иногда его так и не удастся найти), другие же, напротив, столь просты и привычны, что решаются автоматически. При этом выполнение даже самой простой задачи осуществляется в несколько последовательных этапов (шагов). В виде последовательности шагов

можно описать процесс решения многих задач, известных из школьного курса математики: приведение дробей к общему знаменателю, решение системы линейных уравнений путем последовательного исключения неизвестных, построение треугольника по трем сторонам с помощью циркуля и линейки и т.д. Такая последовательность шагов в решении задачи называется алгоритмом. Каждое отдельное действие – это шаг алгоритма. Последовательность шагов алгоритма строго фиксирована, т.е. шаги должны быть упорядоченными.

Понятие алгоритма близко к другим понятиям, таким, как **метод** (метод Гаусса решения систем линейных уравнений), **способ** (способ построения треугольника по трем сторонам с помощью циркуля и линейки).

1.3. Свойства алгоритмов

Сформулируем основные особенности алгоритмов:

Наличие исходных данных и некоторого результата. Алгоритм – это точно определенная инструкция, последовательно применяя которую к исходным данным, можно получить решение задачи. Для каждого алгоритма есть некоторое множество объектов, допустимых в качестве исходных данных. Например, в алгоритме деления вещественных чисел делимое может быть любым, а делитель не может быть равен нулю.

Массовость, т.е. возможность применять многократно один и тот же алгоритм. Алгоритм служит, как правило, для решения не одной конкретной задачи, а некоторого класса задач. Так, алгоритм сложения применим к любой паре натуральных чисел.

Детерминированность. При применении алгоритма к одним и тем же исходным данным *должен получаться всегда один и тот же результат*, поэтому, например, процесс преобразования информации, в котором участвует бросание монеты, не является детерминированным и не может быть назван алгоритмом.

Результативность. Выполнение алгоритма должно обязательно приводить к его завершению (результату). В то же время можно привести примеры формально бесконечных алгоритмов, широко применяемых на практике. Например, алгоритм работы системы сбора метеорологических данных состоит в непрерывном повторении

последовательности действий («измерить температуру воздуха», «определить атмосферное давление»), выполняемых с определенной частотой (через минуту, час) во все время существования данной системы.

Определенность. На каждом шаге алгоритма у исполнителя должно быть достаточно информации, чтобы его выполнить. Кроме того, исполнителю нужно четко знать, каким образом он выполняется. Шаги инструкции должны быть достаточно простыми, элементарными, а исполнитель должен однозначно понимать смысл каждого шага последовательности действий, составляющих алгоритм (при вычислении площади прямоугольника любому исполнителю нужно уметь умножать и трактовать знак «х» именно как умножение).

В качестве примера можно рассмотреть алгоритм нахождения середины отрезка при помощи циркуля и линейки:

- поставить ножку циркуля в точку А;
- установить раствор циркуля равным длине отрезка АВ;
- провести окружность;
- поставить ножку циркуля в точку В;
- провести окружность;
- через точки пересечения окружностей провести прямую;
- отметить точку пересечения этой прямой с отрезком АВ.

Основные способы задания алгоритмов:

- ✓ *словесный*, (недостаток – многословность, возможна неоднозначность – «он встретил ее на поле с цветами»),
- ✓ *табличный* (физика, химия и т.д.),
- ✓ *графический* (блок-схемы).

Для разработки и редактирования блок-схем удобно пользоваться программой <http://underlamp.com/projects/bdrf.zip>.

1.4. Исполнители для школьников

Роль исполнителей как оптимального средства формирования умения алгоритмизации сформулирована еще С. Пайпертом и на сегодня общепризнана. Так, еще в старых версиях Бейсиках появился исполнитель над графикой на основе команды DRAW, а в «австралийской» версии Паскаля для IBM-360 (1980 г.) – версия «черепашки» на текстовом экране. Позже наступило время «чистых» исполнителей и учебных алгоритмических языков над ними. Выбор языка для реализации исполнителя определялся возможностями ЭВМ и вкусом разработчика. Характерно, что язык реализации обычно тщательно скрывается разработчиком от обучаемого, но различие

малых и больших букв выдают влияние языка Си в «КуМире», а многочисленные окошки и картинки в ИнтАле – среду разработки типа DELPHI.

Должен ли эволюционировать такой исполнитель вслед за языком разработки? С точки зрения **цели разработки** – основ алгоритмизации – **не должен(!)**, как не эволюционирует букварь или таблица умножения. Поэтому не меняются десятилетиями ни КуМир, ни ИнтАл, как не меняется **их цель – обучение алгоритмизации на исполнителях.**

Но все же языки программирования развиваются, рождаются новые языки и версии. И все это происходит довольно быстро. Возникает опасность разрыва между растущими языками как средством написания программ и более консервативными версиями УАЯ со встроенными исполнителями. Для уменьшения этого разрыва возможны решения:

1) можно дополнять стандартных исполнителей адаптированными возможностями растущих языков. Это решение принято разработчиками семейства LOGO. Черепашек стало много (аналогия параллельных процессов), и они начали взаимодействовать (обработка событий).

2) можно приблизить УАЯ (на примере ИнтАла) к следующему изучаемому языку (здесь имеется в виду Паскаль). Следующий радикальный шаг в этом направлении – украинская трехязычная версия Паскаля с исполнителями: ученику переводить с английского вообще ничего не надо;

3) наконец, можно реализовать исполнителя как открытую для обучаемого среду и команды-подпрограммы в среде изучаемого языка. Примером служит созданный исполнитель «Робот*» (см. Глава 3 данного издания и [17]). Другие подобные исполнители авторам не известны.

При последнем подходе УАЯ в общем должны оставаться тем, чем они являются – средством обучения алгоритмизации. Ступенькой от алгоритмизации к программированию на конкретном языке становится реализация уже знакомого по УАЯ или похожего на него исполнителя средствами изучаемого языка и управление им прямо из этого языка. Целью здесь является ответы на вопросы: «Как устроен сам исполнитель?», «Как его программировать?», и затем «Как далее программировать в этом языке другие задачи?».

Глава 2. Формирование алгоритмического мышления

2.1. Комплексы задач разного уровня сложности для формирования понятия «событие» и алгоритмического стиля мышления школьников

Приведем комплексы задач разного уровня сложности, которые могут оказать помощь в подготовке к проведению факультативных занятий по информатике.

Знакомство с понятием «алгоритма» обычно начинается с задачи о **Перевозке**. Далее идет целый ряд классических алгоритмических этюдов, в том числе и о переливаниях. Задачи представляют из себя цепочки – от простейших к более сложным. В эти комплексы вошли задачи из различных сборников [3, 4, 5], а также авторские. Наиболее полно решения задач приведены в [3, 6].

Перевозчик

Задачи на перевозку и переправы разделены на несколько групп по возрастанию сложности:

- ✓ рассматривается разное количество героев, участвующих в переправе;
- ✓ и различная вместимость лодки.

При этом условная сложность отдельной задачи внутри той или иной группы оценивается количеством действий, необходимых для решения задачи.

Переправы без условий (событий).

Переправляющиеся находятся на одном берегу.

Начальный этап, стандартные формулировки задач.

- Двухместная лодка: перевоз двух, трех, четырех героев.
- Трехместная лодка: перевоз трех, четырех, пятерых, шестерых героев.

Примеры задач

1. Три поросенка, три братца, пошли погулять. На их пути встретилась глубокая канава. Через нее была положена доска, по ней можно переправляться вдвоем. Однако после переправы двоих поросят останется один поросенок, которому придется переходить мостик в одиночку. А Нуф-Нуф и Ниф-Ниф категорично отказались идти по одному – они боялись потерять равновесие и упасть в канаву. Наф-Наф оказался смелее и сказал, что он всем поможет. Что придумал Наф-Наф?

2. Утке Крякуше подложили куриные яйца, и она высидела пятерых утят и четырех цыплят. Однажды все семейство отправилось гулять на речку. На другом берегу зеленела аппетитная трава, а в воде у другого берега было большое количество ряски. Утята с радостью бросились в воду и переплыли речку. Мама утка умела перевозить на спине двоих цыплят. Как она перевезла всех четверых?

3. В другой раз цыплята и утята играли во дворе в искусственном водоеме. Цыплята нашли маленькую игрушечную лодочку и решили повторить переправу через речку без мамы утки. Цыпленок Командир сказал, что он будет Крякушей и всех цыплят перевезет, только не сам, как мама, а с помощью двухместной лодочки. Сколько раз ему пришлось переплывать с берега на берег?

4. Семеро гномов возвращались домой. И вдруг дорогу перегородила огромная лужа. Она была не очень широкой, но очень длинной, не было видно ее конца ни слева, ни справа. Гномы увидели кусок коры и быстро соорудили подобие лодочки. В ней могло поместиться трое. Как быстрее всего можно им переправиться на другой берег?

Переправы без условий.

Переправляющиеся находятся на разных берегах.

Начальный этап, стандартные формулировки задач.

Одноместная лодка.

- Переезжают три героя.
На первом берегу два героя, на втором – один.
- Переезжают пять героев.
На первом берегу три героя, на втором – два.
- Переезжают семь героев.
На первом берегу четыре героя, на втором – три.

Примеры задач

5. Ежик нашел в лесу грибы и решил отнести их белке. Белка заготовила для ежика на угощение лесные яблоки. На спине у ежа помещается один грибок или два яблока. Если грибов 2 (3, 4) и яблок 2 (3, 4), то сколько раз будет ходить ежик к белке и обратно?

6. Дорога из деревни Отмичи в деревню Кокошки пересекает реку Тьму (названия можно менять на местные). Обычно речку переходили вброд, а в один год лето было особенно дождливым и для переправы приспособили одноместную лодку. Как-то раз к реке подошли 3 (4) человек с первого берега и 2 (3) – со второго. Можно ли

всем переправиться? За сколько переездов состоится переправа пешеходов?

7. Во дворе играли четверо детей. К ним присоединился пятый. У него был велосипед (самокат). Всем очень хотелось покататься на нем. Дети заспорили и никак не могли разобраться, кто поедет раньше. Тут подошла мама одного мальчика и организовала игру. Дети разбились на две группы и встали на расстоянии. Группам надо поменяться местами с помощью велосипеда (самоката). Игра повторяется несколько раз на скорость и определяется рекордное (самое короткое) время. Как могли разделиться на группы дети, у какой группы был вначале велосипед (самокат) и за сколько переездов игра заканчивается?

Двухместная лодка (есть событие – лодка наполнена).

Начальный этап, стандартные формулировки задач.

- Переезжает четыре человека.
На первом берегу – три героя, на втором – один.
- Переезжает четыре человека.
На первом берегу – один герой, на втором – три.
- Переезжает пять человек.
На первом берегу – четыре героя, на втором – один.
- Переезжает пять человек.
На первом берегу – три героя, на втором – два.
- Переезжает пять человек.
На первом берегу – два героя, на втором – три.
- Переезжает пять человек.
На первом берегу – один герой, на втором – четыре.

Примеры задач

8. В зимние каникулы ребята из соседних деревень отправились в гости к бабушкам. В деревне Отмичи гостит трое ребят, а в деревне Кокошки – один. Но в последний день снегу выпало так много, что вернуться домой можно было только на санях (снегоходе). Сани (снегоход) находятся в Отмичах. Управляет ими взрослый, а еще уместиться может только двое детей. За сколько поездок все ребята вернуться домой?

9. Как изменится решение задачи, если в Отмичах гостил один, а в Кокошках – трое ребят?

10. Муравей перетаскивает зернышки из одного домика в другой. Одновременно он может нести два зернышка. В первом домике

лежит два (одно) зернышка, а во втором – три (четыре). За сколько переходов он выполнит свою работу?

Трехместная лодка.

Начальный этап, стандартные формулировки задач.

- Переезжает пять человек.
На первом берегу – четыре героя, на втором – один.
- Переезжает пять человек.
На первом берегу – три героя, на втором – два.

Примеры задач

11. В конце учебного года необходимо сдать старые учебники в школьную библиотеку и получить новые. Эта работа была поручена пяти ученикам. Один ученик может одновременно взять три комплекта. Если в классе 30 учеников, то за сколько походов в библиотеку управятся дети?

12. В лифт вмещается три человека. Лифт имеет особенность: он управляется только из кабины и перемещается только между двумя этажами: первым и последним. На первом этаже находится 4 человека, которым надо подняться на последний этаж, а на последнем – 1, который ждет спуска. За сколько поездок все люди смогут переместиться? Как изменится решение, если внизу находится 1 человек, а сверху – 4? Начальное положение лифта на первом этаже.

13. На острове устроили лагерь для туристов. Наступил конец одной смены и начало другой. В распоряжении лагеря имеется трехместная лодка. Каждая смена состояла из 9 (12, 15, 18) туристов. Как быстрее всего вывезти одних туристов с острова, а других перевезти на остров?

Переправы с условиями. Условия на вместимость лодки.

Начальный этап, стандартные формулировки задач.

Папа с двумя сыновьями отправился в поход. На их пути встретилась река. У берега – плот. Он выдерживает на воде только папу или двух сыновей. Как переправиться на другой берег папе и сыновьям?

Примеры задач

14. Два (три) спортсмена-байдарочника подошли к переправе, около которой они увидели лодку и двух мальчиков. В лодке может плыть

один взрослый или два мальчика. За сколько переездов спортсмены переправятся на другой берег?

15. Спортсмены-байдарочники (из предыдущей задачи) хорошо плавают, поэтому часть из них решила речку переплыть самостоятельно, без лодки. Скорость спортсмена в 2 (3, 4) раза меньше скорости лодки. Скорость лодки одинакова, кто бы в ней не плыл. Как организовать самую быструю переправу? Скольким спортсменам лучше переплыть реку и скольким имеет смысл плыть на лодке? При одинаковом времени переправы предпочтительнее максимальное количество перевезенных на лодке, т.е. оставшихся “сухими”.

16. Коза и семеро козлят пошли в лес за ягодами. На пути им пришлось перебираться на лодке через реку. Лодка вмещает либо одну козу, либо двух козлят. Если один переезд занимает 3 минуты, то сколько времени будет длиться переправа?

17. На городском туристическом слете во время соревнований устроили необычную переправу. Команде из 8 человек надо переплыть на другой берег реки. В их распоряжении имеется две лодки и четыре помощника. Лодка вмещает либо одного туриста, либо двух помощников. Помогите туристам правильно (быстрее всего) переправиться на другой берег.

Переливашка

Задачи на деление некоторого количества жидкости с помощью двух дополнительных пустых сосудов за наименьшее число переливаний

В качестве событий можно выбрать «ведро полно» (из задачи о наполнении 7-литрового ведра) или «необходимый уровень уже есть» (в задачах на точный уровень воды).

Рассматриваются различные объемы сосудов и получение разного количества жидкости. Сложность отдельной задачи оценивается количеством переливаний (ходов), необходимых для решения.

Распределение задач по сложности (в зависимости от сложности – количества ходов – приведены объемы жидкости, которые можно получить путем переливаний с помощью трех сосудов заданного объема).

Примеры задач

1. Две группы альпинистов готовятся к восхождению. Для приготовления еды они используют примусы, которые заправляют бензином. В альплагере имеется 10-литровая канистра бензина.

Имеются еще пустые сосуды емкостью в 7 и 2 литра. Как разлить бензин в два сосуда по 5 литров в каждом?

2. Как разделить поровну между двумя семьями 12 литров хлебного кваса, находящегося в 12-литровом сосуде, воспользовавшись для этого двумя пустыми сосудами: 8-литровым и 3-литровым?

3. У Карлсона есть ведро варенья, оно вмещает 7 литров, а также 2 пустых ведерка – 4-литровое и 3-литровое. Помогите ему отлить 1 литр варенья к чаю в меньшее (3-литровое) ведерко, оставив 6 литров в большом (7-литровом) ведре.

4. Летом Вини-Пух сделал запас меда на зиму и решил разделить его пополам, чтобы съесть половину до Нового года, а другую половину – после Нового года. Весь мед находится в ведре, которое вмещает 6 литров, у него есть 2 пустые банки – 5-литровая и 1-литровая. Может ли он разделить мед так, как задумал?

5. На другой год Вини-Пух запасся 10 литрами меда. Под руками у него два ведра – 7-литровое и 4-литровое. Как ему разделить мед пополам?

6. (Пересыпашка). Разбойники раздобыли 10 унций (1 унция – примерно 30 см^3) золотого песка. У них имеется две пустые коробки, емкостью 6 и 4 унции. Как им разделить песок пополам? Если на одно пересыпание требуется 1 минута, то сколько времени они будут делить свою добычу?

7. Некто имеет полный бочонок сока емкостью 12 пинт (пинта – 0,57 литра) и хочет подарить половину своему другу. Но у него нет сосуда в 6 пинт, а есть два сосуда в 8 пинт и 5. Каким образом можно налить 6 пинт в сосуд емкостью 8 пинт?

8. Белоснежка ждет в гости гномов. Зима выдалась морозной и снежной, и Белоснежка наверняка не знает, сколько гномов решатся отправиться в далекое путешествие в гости, однако знает, что их будет не более 12. В ее хозяйстве есть кастрюлька на 12 чашек, она наполнена водой, и две пустые – на 9 чашек и на 5. Можно ли приготовить кофе для любого количества гостей, если угощать каждого одной чашкой напитка?

9. Разрешима ли предыдущая задача, если в хозяйстве у Белоснежки имеются кастрюлька с водой на 12 чашек и пустые кастрюльки на 9 и 7 чашек?

10. Для путешествия по морю необходим запас пресной воды. В плавании вода расходуется со скоростью 1 бочка в сутки. В некоторый момент времени запас воды на берегу составлял 8 бочек, и вода находилась в баке, заполненном до краев. На яхте имеется такой же бак, объемом 8 бочек, но пустой. На сколько дней можно планировать путешествие, если с собой нельзя брать лишнюю воду, а

в распоряжении имеется еще две пустые емкости объемом 3 и 6 бочек и их можно использовать для переливания воды?

Задачи на получение некоторого количества жидкости из большого или бесконечного по объему сосуда, водоема или источника с помощью двух пустых сосудов

Примеры задач

1. Для разведения картофельного пюре быстрого приготовления “Зеленый великан” требуется 1 л воды. Как, имея два сосуда емкостью 5 и 9 литров, налить 1 литр воды из водопроводного крана?
2. Для марш-броска по пустыне путешественнику необходимо иметь 4 литра воды. Больше он взять не может. На базе, где имеется источник воды, выдают только 5-литровые фляги, а также имеются 3-литровые банки. Как с помощью одной фляги и одной банки набрать 4 литра во флягу?
3. Приготовили ведро компота. Как, имея банки, вмещающие 500 г и 900 г воды, отливать компот порциями по 300 г?
4. Нефтяники пробурили скважину нефти. Необходимо доставить в лабораторию на экспертизу 6 литров нефти. В распоряжении имеется 9-литровый и 4-литровый сосуда. Как с помощью этих сосудов набрать 6 литров?
5. Как решить предыдущую задачу, если на экспертизу необходимо доставить 5 литров нефти, а емкости сосудов составляют соответственно 7 литров и 3?

Глава 3. Исполнитель, управляемый событиями

3.1. Основные результаты. Функциональные возможности исполнителя

Рабочая программа представляет собой файл **Робот.xls** [17]. Для начала работы с исполнителем необходимо открыть этот файл в любой версии Excel, предварительно разрешив выполнение макросов. Общий вид экрана после запуска программы показан на рис. 1.

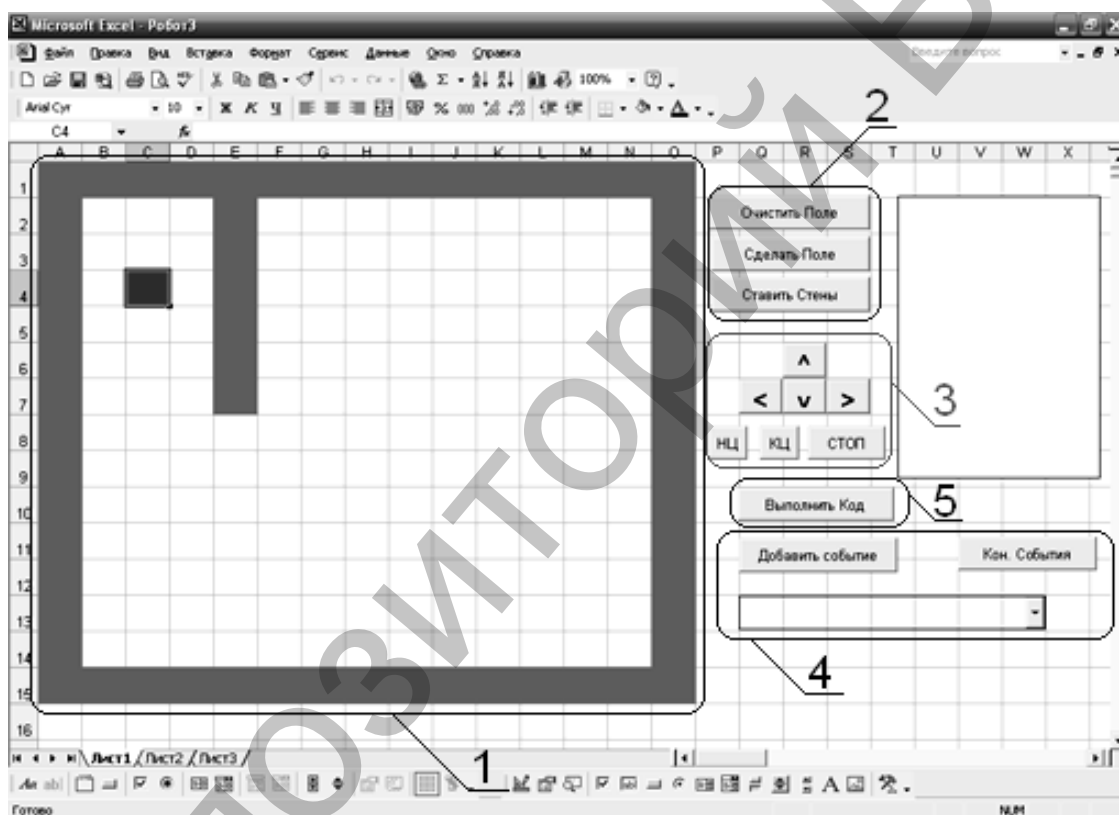


Рис. 1. Общий вид программы после старта.

Цифрами на рисунке обозначены следующие элементы:

1. **Стандартное поле исполнителя:** в программе предусмотрена возможность произвольного задания начального поля.

2. **Опции поля:** в помощью этой панели пользователь может очистить поле, создать поле (загрузится стандартное поле) и расставить или убрать стены на готовом поле.

3. **Кнопки управления исполнителем:** с помощью этих кнопок пользователь вводит все необходимые команды для исполнителя: «влево», «вверх», «вниз», «вправо», «нц», «кц», «стоп».

4. **Область ввода событий:** В разворачивающемся списке представлены все события доступные исполнителю.

5. **«Выполнить Код»** – с помощью этой кнопки программа запускается на выполнение.

В предложенном исполнителе **«Робот»** запрограммированы следующие команды: «влево», «вправо», «вверх», «вниз», «стоп», парный оператор «нц–кц» и список событий: "Слева свободно", "Сверху свободно", "Справа свободно", "Снизу свободно", "Слева стена", "Сверху стена", "Справа стена", "Снизу стена", "On Error (любая ошибка)", что является несомненным расширением стандартных возможностей исполнителей из среды КуМир и ИнтАЛ.

Как видно из набора допустимых команд, представленный исполнитель полностью обходится без условий в традиционном смысле: "нц пока", "если", "при". Обоснование такого подхода описано в статье [2].

Команды «влево», «вправо», «вверх», «вниз» определяют перемещение робота по горизонтали и вертикали соответственно, парный оператор «нц–кц» представляет собой бесконечный цикл, выход из которого возможен только при наступлении события (см. Пример 1).

Пример 1

при Справа стена
стоп
конец события
нц
вправо
кц

В рассмотренном примере команда «вправо» будет выполняться до тех пор, пока справа будет свободно (*наступление события проверяется один раз перед началом выполнения основной программы, а потом – после выполнения каждой команды!*), но как только справа будет стена (*наступило событие «Справа стена»*) выполнение программы будет остановлено командой «стоп» события.

Событие «On Error (любая ошибка)» носит более общий характер, чем остальные – оно дает пользователю возможность запрограммировать действия исполнителя в случае попытки

выполнить любое недопустимое действие (например, если нам неизвестно, в каком месте будет стена).

Таким образом, с помощью представленного исполнителя можно объединить базовые понятия объектно-ориентированного программирования и УАЯ при изучении школьного курса алгоритмизации.

Управление полем исполнителя осуществляется с помощью трех кнопок панели настройки (рис. 2).

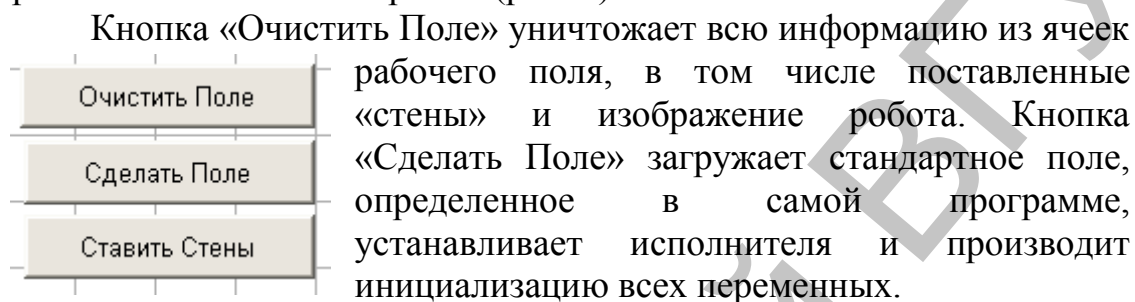


Рис. 2.

Для установки стен в произвольном месте поля необходимо выделить нужную область и нажать кнопку «Ставить Стены», для удаления стен нужно **выделить область, содержащую ненужные элементы**, и снова нажать «Ставить Стены».

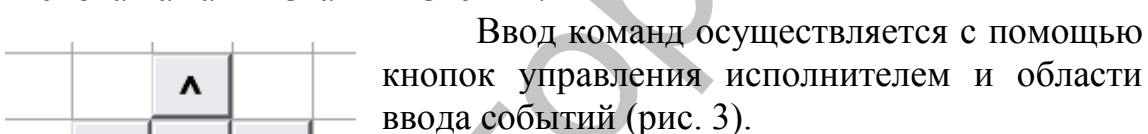


Рис. 3.

Итак, предлагаемый исполнитель управляется исключительно событиями и включает в себя все команды стандартных исполнителей из КуМира и ИнтАла. Конечно, он не отменяет работу учащихся с привычными исполнителями, однако связывает важную часть современной информатики – программирование событий и прерываний со школьной программой. Такой исполнитель может изучаться даже до стандартных средств алгоритмизации.

Глава 4. Программная реализация исполнителя

Вся программа разбита на три модуля, в которых сосредоточены основные процедуры, Лист 1 представляет собой рабочую среду, Лист 2 – краткое справочное руководство (полные листинги процедур приведены в [17]).

Разбиение программы на модули выполнено в методических целях: наиболее простые процедуры расположены в первом модуле, более сложные – в последующих, что позволяет каждый модуль в отдельности использовать на разных стадиях обучения алгоритмизации. Каждая процедура модуля снабжена необходимыми комментариями, что избавляет учителя от необходимости устно комментировать весь код и дает возможность учащимся самостоятельно работать с исполнителем.

4.1. Основные глобальные переменные

В начале первого модуля представлены основные глобальные переменные, используемые в программе (рис. 4.1).

```
Public RX, RY
Public point
Public aktivn_ysl ' 0..8 - активное условие для события
Public ev_point 'указатель для массива с событием

Public ev_code(20) As String 'массив команд очередного события
Public code(100) As String ' массив команд исполнителя
Public evMass(8) As Integer ' константы поля (слева_свободно=да, ...) yes=1, no = 0
Public Start_point ' переменные для циклов нц-кц
Public Global_Stop 'останов выполнения программы
Public Stop_point ' указатель на номер последней команды программы
```

Рис. 4.1. Глобальные переменные.

RX и **RY** – текущие координаты робота на поле, **code** – массив, в котором содержатся все введенные пользователем команды для «Робота», **point** – указатель на текущую команду в этом массиве. Рассмотрим основные процедуры программы.

4.2. Процедура *Make_field()*

Эта процедура вызывается нажатием кнопки «Создать Поле» рабочей области. В начале процедуры происходит установка начальных значений глобальных переменных, выравнивание рабочей области, устанавливаются начальные координаты робота. Далее, для формирования начального поля (рис. 4.2), вызывается процедура *Walls()* – устанавливает стену в заданной ячейке.

Заменяв предложенный в программе код, можно смоделировать поле для любой задачи из [14] или [15] и загружать его по умолчанию (например, для моделирования возможных ситуаций задачи (в теме «Лабиринты», или сравнения полученных решений задач на факультативе).

```

For j = 1 To 15
  Call Walls(1, j)
  Call Walls(15, j)
Next j

For j = 2 To 14
  Call Walls(j, 1)
  Call Walls(j, 15)
Next j

For j = 2 To 7
  Call Walls(j, 5)
  Call Walls(j, 5)
Next j

```

Далее в процедуре организовано заполнение раскрывающегося списка для выбора условия наступления события (рис. 4.3). Список условий не является жестким и при необходимости его можно легко расширить для решения нового класса задач.

Рис. 4.2.

```

' ----- делаем ComboBox для списка условий
With ActiveSheet
  .ComboBox1.AddItem ("Слева свободно")
  .ComboBox1.AddItem ("Сверху свободно")
  .ComboBox1.AddItem ("Справа свободно")
  .ComboBox1.AddItem ("Снизу свободно")
  .ComboBox1.AddItem ("Слева стена")
  .ComboBox1.AddItem ("Сверху стена")
  .ComboBox1.AddItem ("Справа стена")
  .ComboBox1.AddItem ("Снизу стена")
  .ComboBox1.AddItem ("On Error (любая ошибка)")
End With

```

Рис. 4.3. Создание списка событий.

4.3. Процедура Exec_ev()

Данная процедура (рис. 4.4) запускает выполнение заданного кода при наступлении события. В первой части процедуры проверяется условие наступления события (*aktivn_ysl*) и, если оно истинно ($evMass(aktivn_ysl) = 1$), то выполняем предписанные действия.

Во второй части мы проверяем еще раз условие события (оно могло стать ложным после действий в первой части. Например, робот шагнул вправо и «уперся» в стену, в таком случае событие «Справа свободно» уже станет ложным) и, если оно истинно, то процедура рекурсивно вызывает себя еще раз, в противном случае завершает свою работу.

```

Sub Exec_ev()
'
' выполняем команды события
'
ev_point = 1
If (evMass(aktivn_ysl) = 1) Then
While (ev_code(ev_point) <> "") And (Not (Global_Stop))
Call Module3.move(ev_code(ev_point)) ' -- двигаем робота
ev_point = ev_point + 1
Wend
End If

Refresh_EventMass ' обновим массивы констант поля
If (evMass(aktivn_ysl) = 1) And (Not (Global_Stop)) Then
Call Exec_ev '-- еще раз себя вызовем
End If

End Sub

```

Рис. 4.4. Процедура обработки события.

4.4. Процедура move()

Данная процедура обеспечивает перемещение робота по клеточному полю. Полный листинг этой процедуры представлен в Приложении.

В качестве параметра подпрограмме передается команда в виде

```

Cells(RX, RY).Select
If Selection.Interior.ColorIndex = 16 Then
Call Error(5) 'ошибка! - здесь стена!!
RX = OldX
RY = OldY
Cells(RX, RY).Select
Else
With Selection.Interior
.ColorIndex = 10
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
End With
End If

```

текстовой строки. После анализа этой команды в процедуре принимается решение о перемещении робота: **сместиться по команде (если это возможно)** или остаться на месте и **вывести сообщение об ошибке (например, врезались в стену)** рис. 4.5.

Рис. 4.5.

Затем проверяем условие наступления **активного** события и,

если оно выполнено, то вызываем соответствующую процедуру обработки (рис. 4.6).

```

' ----- анализ сделанного шага
Refresh_EventMass ' обновим массивы констант поля
If evMass(aktivn_ysl) = 1 Then
Call Exec_ev '-- вызовем обработчик события
End If

```

Рис. 4.6. Анализ сделанного шага.

4.5 Выполнение кода

После нажатия кнопки «Выполнить Код» запускается специальная подпрограмма-обработчик. Процедура анализирует каждую строку из массива команд и принимает необходимое решение.

Рассмотрим алгоритм обработки более подробно:

если в начале анализируемой строки встретилось ключевое слово «*при*» (с этого оператора начинается описание события), то в процедуре вырезается из этой строки условие и устанавливается соответствующее значение переменной **aktivn_ysl**. Далее, пока не будет достигнут код конца события, считываются все команды в массив **ev_code** и запускается процедура обработки события (рис. 4.7).

```
If InStr(code(point), "при") > 0 Then
' --- перед нами событие !!!!!
' надо прочитать все событие в массив ev_code и вызвать Exec_ev
' предварительно установив aktivn_ysl
s$ = Mid$(code(point), 5, Len(code(point)))
Select Case s$ ' устанавливаем условие, на которое нужно обратить внимание
Case "Слева свободно"
aktivn_ysl = 1
Case "Сверху свободно"
aktivn_ysl = 2
Case "Справа свободно"
aktivn_ysl = 3
Case "Снизу свободно"
aktivn_ysl = 4
Case "Слева стена"
aktivn_ysl = 5
Case "Сверху стена"
aktivn_ysl = 6
Case "Справа стена"
aktivn_ysl = 7
Case "Снизу стена"
aktivn_ysl = 8
Case Else
aktivn_ysl = 0 'выбрно "любая ошибка исполнителя"
End Select ' <===== установили активное условие для ЭТОГО события

point = point + 1 ' смещаемся к следующему элементу
ev_point = 1 ' ставим начальное значение для НОВОГО события
While code(point) <> "конец события"
ev_code(ev_point) = code(point)
point = point + 1
ev_point = ev_point + 1
Wend
Call Module2.Exec_ev ' вызываем обработчик прочитанного события
```

Рис. 4.7. Обработка события.

Если в строке встречается оператор «*нц*» (нужно обработать цикл «*нц*»—«*кц*»), то выполняются все команды до строки, содержащей «*кц*», а затем указатель номера выполняемой команды возвращается в начало цикла. В данном случае выход из такого бесконечного цикла возможен только по событию (рис. 4.8).

В противном случае, если перед нами «обычная» команда, то запускается процедура **move()** и в качестве параметра ей передается эта команда.

```
While (Not (Global_Stop))
While (code(point) <> "КЦ") And (Not (Global_Stop))
Call Module3.move(code(point)) ' -- двигаемся, куда указано
point = point + 1
Wend ' нашли конец тела цикла
point = Start_point ' возвращаемся к началу тела цикла ^^^
Wend
```

Рис. 4.8. Обработка цикла.

Приложение А. План-конспект урока

Тема: «Алгоритм и его свойства»

Общие сведения:

- раздел «Алгоритмизация и программирование»
- учебная тема «Алгоритм и его свойства»
- номер урока – 1, 2.
- учебные часы – 2.

Вводно-мотивационный этап

Цели

- образовательная: сформировать представление у учащихся о понятии «алгоритма», выделить его свойства. Рассмотреть понятие «исполнителя» и «системы команд исполнителя»;
- развивающая: формирование приемов логического мышления, развитие интереса к предмету;
- воспитательная: воспитание аккуратности, точности.

Ядро содержания обучения: понятия «алгоритма», «исполнителя» и «системы команд исполнителя».

Предварительная подготовка учащихся: изучение конкретных алгоритмов и их исполнителей на других предметах, изученный материал на предыдущих уроках информатики.

Предварительная подготовка учителя: изучение материала урока, написание конспекта, разработка методических пособий.

Дидактические основания урока:

- *методы обучения:* объяснительно-иллюстративный;
- *тип урока:* комбинированный;
- *формы учебной работы учащихся:* фронтальная работа, работа в парах.

План урока:

1. Организационный момент (1–2 мин).
2. Актуализация опорных знаний (5 мин).
3. Объяснение нового материала (35 мин).
4. Решение задач (35).
5. Итог урока (3 мин).

1. Содержательно-деятельностный компонент (ход урока)

Деятельность учителя	Деятельность учащихся
Здравствуйте, садитесь. Запишем тему урока: «Алгоритм и его свойства».	«Алгоритм и его свойства».
В своей жизни мы встречаемся с различными практическими задачами: например, приготовление супа, решение уравнения, покупка продуктов и т.д. При решении любой задачи человек выполняет некоторую последовательность действий. Давайте приведем пример.	Слушают.
Какую последовательность действий нужно выполнить, чтобы позвонить по телефону-автомату?	<ol style="list-style-type: none">1. Вставить телефонную карточку.2. Снять трубку.3. Набрать номер.
Хорошо, эту последовательность действий может выполнить каждый из вас. Так вот, эта последовательность и называется алгоритмом. Давайте запишем определение: (учитель диктует определение).	<i>Алгоритм – это точное предписание о последовательности действий, которые должны быть произведены для получения результата.</i>
Давайте вспомним, где вы уже встречались с алгоритмами?	На информатике (этапы решения задач), на математике (решение уравнения) и т.д.
Хорошо, а теперь запишем следующее определение:	<i>Исполнитель – объект, который выполняет алгоритм.</i>

<p>А теперь подумайте: приведенные вами примеры алгоритмов сможет выполнить любой человек?</p>	<p>Нет, алгоритм решения квадратного уравнения не сможет выполнить младший школьник.</p>
<p>То есть, какой можно сделать вывод?</p>	<p>Алгоритм может выполнить тот, кто понимает все его команды и может их выполнить.</p>
<p>Таким образом, мы видим, что алгоритм не имеет смысла, если неизвестны или не учитываются возможности того, кто будет исполнять этот алгоритм, то есть возможности исполнителя. Поэтому нам потребуется еще одно определение «Система Команд Исполнителя».</p>	<p>Слушают.</p>
<p>Запишем, что же это такое:</p>	<p><i>Система команд исполнителя – совокупность команд, которые данный исполнитель умеет выполнить.</i></p>
<p>Как вы думаете, кто может служить исполнителем алгоритма?</p>	<p>Человек, автомат, компьютер, машина и т.д.</p>
<p>Как вы поняли, каждый алгоритм должен быть понятен исполнителю, поэтому алгоритм должен быть записан на понятном для исполнителя языке, и эта запись называется программой. Запишем:</p>	<p><i>Программа – запись алгоритма на языке исполнителя.</i></p>
<p>Итак, мы выяснили, что такое алгоритм и его исполнитель. Давайте теперь подумаем, какими свойствами обладает алгоритм.</p>	<p>Должны быть отдельные шаги.</p>
<p>Хорошо, давайте запишем: (учитель диктует).</p>	<p><i>Свойства: дискретность – процесс решения задачи должен быть разбит на последовательность отдельных шагов.</i></p>

<p>Кроме того, исполнитель переходит к выполнению следующей команды, только выполнив предыдущую. Как вы думаете, каким еще должен быть алгоритм? Запишем:</p>	<p>Понятным исполнителю. <i>Понятность. Алгоритм должен быть понятен исполнителю и исполнитель должен быть в состоянии выполнить его команды.</i></p>
<p>Хорошо ясно, что алгоритм составляется с учетом возможностей исполнителя. Следующее свойство алгоритма называется свойством определенности, то есть команда воспринимается однозначно. Например, робот не поймет команды положить 2–3 ложки песка. Запишем:</p>	<p><i>Определенность. Алгоритм не должен содержать команды, смысл которой может восприниматься неоднозначно.</i></p>
<p>Какое свойство еще присуще алгоритму, к чему он всегда должен приводить?</p>	<p>К результату.</p>
<p>Хорошо, это свойство называется результативность. Давайте запишем:</p>	<p><i>Результативность. Процесс решения задачи должен прекратиться за конечное число шагов и при этом должен быть получен ответ задачи.</i></p>
<p>И последнее свойство – это массовость, то есть по одному алгоритму можно решать однотипные задачи. Давайте зафиксируем это свойство в тетради.</p>	<p><i>Массовость. По одному алгоритму можно решать однотипные задачи.</i></p>
<p>Итак, мы рассмотрели свойства алгоритмов, давайте еще раз их перечислим.</p>	<p>Дискретность, понятность, определенность, результативность, массовость.</p>
<p>Хорошо. Все эти свойства, с другой стороны, можно рассматривать как требования к построению алгоритма, т.е. когда будете строить алгоритм решения задачи вы должны следовать этим</p>	<p>С помощью блок-схем, с помощью слов.</p>

<p>требованиям. Хорошо, давайте вспомним: какими способами можно записать алгоритм?</p>	
<p>Совершенно верно. Но, кроме этого, алгоритм можно еще записать на алгоритмическом языке. Давайте запишем:</p>	<p><i>Способы записи алгоритмов:</i></p> <ol style="list-style-type: none"> 1. <i>алгоритмический язык;</i> 2. <i>графический способ (блок-схемы).</i>
<p>Алгоритмы могут отличаться не только по способу записи, но и по виду. Алгоритмы бывают линейные, разветвляющиеся и циклические. Каждый из этих видов на следующих уроках мы подробно рассмотрим, а сейчас просто запишем в тетрадь.</p>	<p><i>Виды алгоритмов:</i></p> <ol style="list-style-type: none"> 1. <i>Линейный – все действия выполняются последовательно.</i> 2. <i>Разветвляющийся – выполнение действий зависит от условий.</i> 3. <i>Циклический – действия повторяются многократно.</i>
<p>Хорошо, а теперь давайте решим одну задачку. Исполнитель умеет заменить в слове ровно одну букву на любую другую, причем при замене должно получиться осмысленное слово. Составьте алгоритм для преобразования слова САД в слово КОТ. Прежде чем решать, подумаем, какая система команд у нашего исполнителя? Молодцы, давайте составлять алгоритм. Чтобы его составить, нужно сначала решить эту задачу.</p>	<p>Заменять в слове одну букву на другую, причем при замене должно получиться осмысленное слово: САД – САМ – СОМ – КОМ – КОТ.</p>
<p>Составим алгоритм.</p>	<ol style="list-style-type: none"> 1. 3М. 2. 2О. 3. 1К. 4. 3Т.

Хорошо, а теперь будем решать задачи.	Решают задачи (1 ученик решает на доске).
А теперь запишем домашнее задание:	

Схема доски

Тема. Алгоритм и его свойства		
Свойства	Виды	Д/з
1. Дискретность. 2. Понятность. 3. Определенность. 4. Результативность. 5. Массовость.	1. Линейный 2. Разветвляющийся. 3. Циклический	

Словарь терминов

1. *Алгоритм* – это точное предписание о последовательности действий, которые должны быть произведены для получения результата.
2. *Программа* – запись алгоритма на языке исполнителя.
3. *Система команд исполнителя* – совокупность команд, которые данный исполнитель умеет выполнить.
4. *Исполнитель* – объект, который выполняет алгоритм.

Дидактические материалы

1. Составить алгоритм для вычисления длины окружности произвольного радиуса.
2. Составить алгоритм и программу для подсчета и вывода на экран значений косинусов углов 300° , 600° , 900° .
3. Составить алгоритм и программу для вычисления площади треугольника, если известны его стороны.
4. Составить алгоритм и программу для вычисления гипотенузы прямоугольного треугольника, если известны его катеты.
5. Составить алгоритм и программу для вычисления периметра пятиугольника, если известны его стороны.

Подведение итогов урока. Задание на дом.

Приложение Б. Дополнительные задачи к уроку

Задача 1.

Четверо студентов занимаются спортом и участвуют в художественной самодеятельности. Определите, кто из них чем увлекается, если известно:

Боксер любит баклажанную икру, а Миша предпочитает кабачковую.

Студент, занимающийся в балетном кружке, дружит со штангистом.

Гриша – большой оригинал.

Студент, занимающийся в кружке художественного свиста, считает, что если бы шахматист был силен, как Тиша, и ловок, как Миша, то он пользовался бы у девушек колоссальным успехом.

Эдуард каждое утро чистит ботинки, чего нельзя сказать, к сожалению, о штангисте.

У студента, занимающегося в кружке художественной вышивки, голубые глаза.

Боксер считает ниже своего достоинства участвовать в балетном кружке. Тиша так не считает, но сам в нем не занимается.

Штангист Тишу не знает и знать не желает.

Боксер любит Машу, Маша любит Эдуарда, а Эдуард поет в хоре.

...а четвертый студент – футболист.

Задача 2. «Задача Льва Толстого про шапку»

Продавец продает шапку. Стоит 10 руб. Подходит покупатель, меряет и согласен взять, но у него есть только банкнота 25 руб. Продавец отсылает мальчика с этими 25 руб. к соседке разменять. Мальчик прибегает и отдает 10+10+5. Продавец отдает шапку и сдачу 15 руб. Через какое-то время приходит соседка и говорит, что 25 руб. фальшивые, требует отдать ей деньги. Ну что делать. Продавец лезет в кассу и возвращает ей деньги.

На сколько обманули продавца?

(По легенде, эта задачка придумана Львом Толстым для второго класса церковно-приходской школы. Сейчас ее правильно могут решить только 30% старшеклассников, только 20% студентов вузов и только 10% работников банков и кредитных учреждений).

Задача 3. «16 команд»

В футбольном чемпионате принимают участие 16 команд, каждая из которых имеет свой стадион. Все команды должны сыграть между собой, при этом в каждом туре проводятся все 8 игр. Можно ли составить расписание туров так, чтоб каждая команда играла попеременно дома и на выезде?

Задача 4. «Что произойдет в городке?»

В деревне, где живет пятьдесят семейных пар, каждый из мужей изменял своей жене. Каждая из женщин в этой деревне, как только кто-то из мужчин изменил своей жене, немедленно узнает об этом (все знают, как быстро распространяются сплетни в маленьких городках), если только это не ее собственный муж (о своих бедах каждый узнает последним).

Законы этого городка требуют, чтобы женщина, получившая доказательства неверности своего мужа, вечером убила его. Ни одна из женщин не может ослушаться. Однажды королева, славящаяся своей непогрешимостью, приезжает в городок. Она объявляет жителям, что по крайней мере один из мужчин городка совершил супружескую измену.

Что произойдет в городке в ближайшее время?

Задача 5. «Загадка Эйнштейна»

5 разных человек в 5 разных домах разного цвета курят 5 разных марок сигарет, выращивают 5 разных видов животных, пьют 5 разных видов напитков.

Норвежец живет в первом доме.

Англичанин живет в красном доме.

Зеленый дом находится слева от белого.

Датчанин пьет чай.

Тот, кто курит Rothmans, живет рядом с тем, кто выращивает кошек.

Тот, кто живет в желтом доме, курит Dunhill.

Немец курит Marlboro.

Тот, кто живет в центре, пьет молоко.

Сосед того, кто курит Rothmans, пьет воду.

Тот, кто курит Pall Mall, выращивает птиц.

Швед выращивает собак.

Норвежец живет рядом с синим домом.

Тот, кто выращивает лошадей, живет в синем доме.
Тот, кто курит Philip Morris, пьет пиво.
В зеленом доме пьют кофе.
Вопрос: кто выращивает рыбок?

По легенде, 98% людей разгадать эту загадку не могут. Это не означает, что те 2%, которые могут, – гении. Они просто умеют решать комбинаторные задачи.

Задача 6. Три ящика

Есть три ящика: ящик с апельсинами, ящик с яблоками и ящик со смесью яблок и апельсинов.

На каждом ящике есть табличка с указанием, что внутри. Таблички взяли и перемешали; теперь все таблички не на своем месте.

Есть одна попытка: можно сунуть руку в ящик и вытащить оттуда 1 предмет.

После этого надо развесить таблички правильно.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Е.А. Пархоменко, Ю.В. Сюбаева, Алгоритм. Способы описания алгоритма: учебно-методическое пособие для учителей информатики / сост. : Лицей № 4, 2005.
2. Бочкин А.И., Кузьмичев Д.Р. «Учебный исполнитель, управляемый событиями» – М.: «Информатизация образования», № 1, 2008 г.
3. Звонкин А.К., Кулаков А.Г., Ландо С.К., Семенов А.Л., Шень А.Х. Алгоритмика. – М.: Дрофа, 1997.
4. Доморяд А.П. Математические игры и развлечения – М.: ГИФМЛ, 1961.
5. Игнатъев Е.И. В царстве смекалки. – М.: Наука, ФизМатЛит, 1987.
6. Первин Ю.А. Алгоритмические этюды, тетрадь № 2. – М.: АО КУДИЦ, 1993.
7. Первин Ю.А. Информатика в школе и дома. – СПб.: БХВ, 2003.
8. Русанов В.Н. Математические олимпиады для младших школьников. – М.: Просвещение, 1990.
9. Шарыгин И.Ф. Математический винегрет. – М.: Издание агентства «Орион», 1991.
10. Шумилина Н.Д. Задачи для «шустриков» и «мямликов» // 3-я научно-методическая телеконференция «Информационные технологии в общеобразовательной школе» (25.11.2002 – 31.03.2003 г.). – Новосибирск, НООС.
11. Шумилина Н.Д. Переправа, переправа (Задачи разного уровня сложности) // Информатика в школе. – 2003. – № 6.
12. Практическая психология образования / под ред. И.В. Дубровиной. – М.: ТЦ «Сфера», 1997.
13. Интернет-ресурс <http://www.1.september.ru>.
14. Кушниренко А.Г., Лебедев Г.В., Сворень Р.А. Основы информатики и вычислительной техники. – М.: Просвещение, 1993.
15. Быкадоров Ю.А., Кузнецов А.Т. Информатика: учеб. пособие для 9-го кл. общеобразоват. шк. – 2-е изд. – Мн.: Нар. света, 1998. – 253 с.: ил.
16. Бочкин А.И., Кузьмичев Д.Р. «О месте оператора GOTO при изучении программирования». «Информатика и образование». – Мн.: №4 за 2008 г.
17. Образовательный интернет-ресурс <http://underlamp.com>.