

Практическое занятие №5.

Составление алгоритмов при помощи базовых алгоритмических конструкций.

Что такое алгоритмическая модель

Попробуем разобраться, почему алгоритм можно назвать моделью и что он моделирует.

Как вам известно, *алгоритм — это понятное и точное предписание конкретному исполнителю совершить конечную последовательность действий, приводящую к поставленной цели.* Из определения следует, что поставленная цель достигается через деятельность (последовательность действий) некоторого исполнителя.

Этапы деятельности от определения цели (постановки задачи) до получения результата такие:

- 1) определение цели;
- 2) планирование работы исполнителя;
- 3) работа исполнителя;
- 3) получение результата.

Где же здесь место алгоритму? Алгоритм — это детальный план работы исполнителя, это описание последовательности элементарных действий, которые должен совершить исполнитель. Но всякий план или описание есть информационная модель. Следовательно:

Алгоритм является информационной моделью деятельности исполнителя.

Такую модель будем называть **алгоритмической**.

В схематическом виде четыре описанных этапа представлены на рис. 3.8.

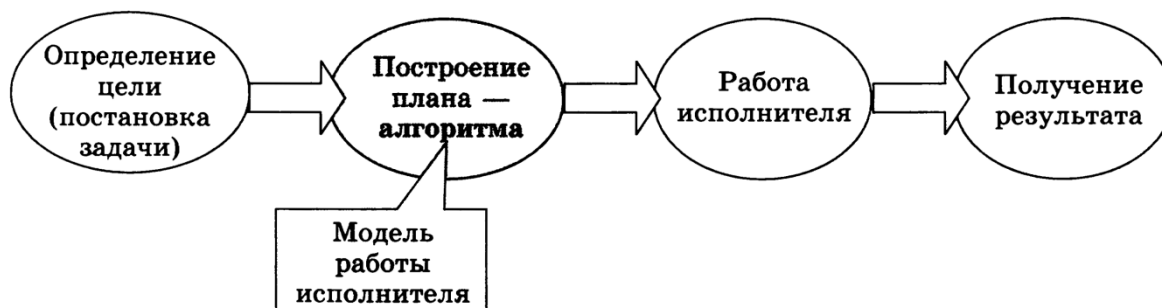


Рис. 3.8. Этапы движения от цели к результату

Чтобы построить реальный план-алгоритм, который окажется выполнимым, нужно точно знать возможности исполнителя. Эти возможности определяются **системой команд исполнителя (СКИ)**. Составляя алгоритм, нельзя выходить за рамки СКИ. В этом состоит свойство понятности алгоритма.

Оказывается, гораздо проще построить алгоритм для программно управляемого автомата (в том числе компьютера), чем для человека. Для автомата СКИ — это строго определенный конечный набор команд, заложенный в него конструкторами. Поэтому алгоритм представляет собой точное описание его работы, и автомат выполняет работу, *формально следуя* указаниям алгоритма. Для управления автоматом или компьютером нетрудно придумать формализованный язык описания алгоритмов. Такие языки называются **языками программирования**, а алгоритм, представленный на языке программирования, называется **программой**.

Сложнее дело обстоит с человеком, которого трудно назвать формальным исполнителем. И что совершенно очевидно, СКИ человека невозможно полностью описать.

Пример алгоритмической модели.

Рассмотрим конкретный пример - угадывание целого числа из заданного диапазона методом половинного деления.

Постановка задачи: первый игрок загадывает целое число из заданного диапазона чисел, например от 1 до 100. Второй должен угадать это число за наименьшее количество вопросов.

Запишем алгоритм угадывания числа методом половинного деления, ориентированный на исполнителя-человека.

Алгоритм Угадывание числа

Дано: диапазон чисел от А до В

Надо: угадать число X, задуманное игроком, используя алгоритм половинного деления

Начало

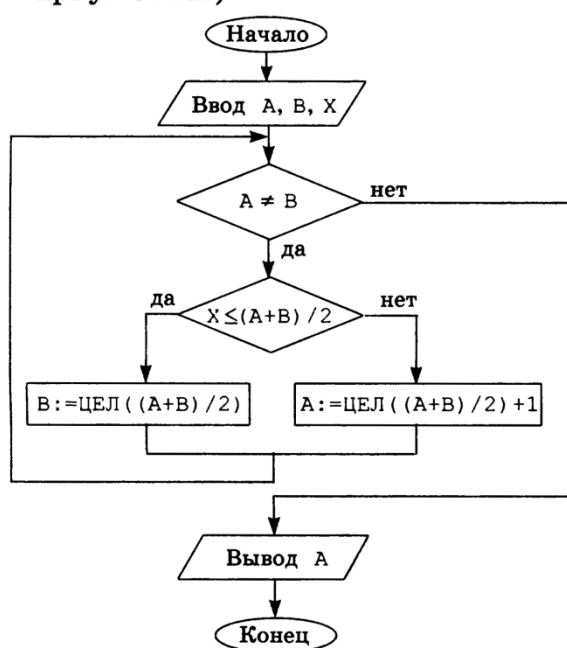
1. Задать вопрос: X меньше среднего значения между А и В?
2. Если ответ "да", то принять за значение В целую часть среднего значения
3. Если ответ "нет", то принять за значение А ближайшее целое число, большее, чем среднее
4. Если значения А и В равны, то их общее значение и есть искоемое число X
5. Если значения А и В не равны, то вернуться к выполнению пункта 1

Конец

Насколько многословен этот алгоритм! И еще нет уверенности, что исполнитель «Вася из 8Б» правильно выполнит все его пункты, хотя образование восьмиклассника должно это ему позволять.

В этом примере использовано словесное описание алгоритма. Данный алгоритм ориентирован на исполнителя-человека, а не на компьютер. Поэтому здесь нет никаких вводов, присваиваний, выводов и прочих формальных команд компьютерного алгоритма. Как уже отмечено выше, один человек его сможет исполнить, а другой — нет.

Алгоритм, составленный для компьютера и переведенный на язык программирования, будет точно исполнен любым компьютером, «понимающим» этот язык. На рис. 3.9 приведен алгоритм поиска числа методом половинного деления для исполнителя-компьютера в форме блок-схемы и на учебном алгоритмическом языке, знакомом вам из базового курса информатики. (Примечание. ЦЕЛ обозначает функцию выделения целой части аргумента.)



Алг Половинное деление

Цел А, В, X

Начало

Ввод А, В, X

Пока А≠В, повторять

Нц

Если X ≤ (A+B) / 2

То В := ЦЕЛ((A+B) / 2)

Иначе А := ЦЕЛ((A+B) / 2) + 1

Кв

Кц

Вывод А

Конец

Рис.3.9 Алгоритм половинного деления

Вспомним основные правила изображения алгоритма с помощью блок-схем.

Блок-схема — это ориентированный граф, указывающий порядок исполнения команд алгоритма исполнителем. Блоки — вершины этого графа — обозначают отдельные команды, которые отдаются исполнителю, а дуги указывают на последовательность переходов от одной команды к другой.

В прямоугольниках на блок-схемах записываются команды — действия, в ромбах — условия, определяющие направление дальнейшего исполнения команд; в параллелограммах — команды ввода или вывода информации; в овалах — начало или конец исполнения алгоритма. Здесь можно говорить о пути прохождения графа в ходе выполнения алгоритма. Любой путь начинается от вершины «Начало» и заканчивается выходом на вершину «Конец». Внутри же путь может быть разным в зависимости от исходных данных и от результатов проверки условий.

Блок-схема и алгоритмический язык — это две разные формы представления алгоритмической модели. Блок-схема — графическая форма, алгоритмический язык — текстовая форма. Блок-схема обладает большей наглядностью, на ней легче увидеть структуру алгоритма. Алгоритмический язык ближе по форме к языкам программирования. От записи алгоритма на алгоритмическом языке легко перейти к записи программы на языке программирования.

Структура построенного алгоритма — *цикл с вложенным ветвлением*. Из базового курса информатики вам должно быть известно, что любой алгоритм можно построить из сочетания трех основных алгоритмических структур: *следования, ветвления и цикла*. Это утверждение — основа методики, которая называется **структурным программированием**. Современные языки программирования позволяют легко переходить от описания алгоритма к программе, если алгоритм построен структурно. Поэтому *наиболее рациональной моделью деятельности исполнителя является структурная алгоритмическая модель*.

Не составит большого труда запрограммировать описанный выше алгоритм на каком-нибудь языке программирования, например на Паскале или Бейсике.

Трассировка алгоритма — модель работы процессора

Для того чтобы проверить правильность алгоритма, изображенного на рис. 3.9, совсем не обязательно переводить его на язык программирования и выполнять тесты на компьютере. Протестировать алгоритм может и человек — путем трассировки. *Выполняя ручную трассировку, человек моделирует работу процессора*, исполняя каждую команду алгоритма и занося результаты выполнения команд в трассировочную таблицу. В базовом курсе вы этим уже занимались. Построим трассировочную таблицу для алгоритма «Половинное деление». Выберем интервал угадываемых чисел от 1 до 8. Пусть игрок задумал число 3. Проверим, как по данному алгоритму будет получено это число.

Трассировочная таблица алгоритма «Половинное деление»

№ шага	Команда алгоритма	Переменные			Выполняемые действия
		X	A	B	
1	Ввод A, B, X	3	1	8	
2	$A \neq B$				$1 \neq 8$, да
3	$X \leq (A+B) / 2$				$3 \leq 4,5$, да
4	$B := \text{ЦЕЛ}((A+B) / 2)$			4	$B := 4$
5	$A \neq B$				$1 \neq 4$, да
6	$X \leq (A+B) / 2$				$3 \leq 2,5$, нет

7	$A := \text{ЦЕЛ}((A+B)/2) + 1$		3		$A := 3$
8	$A \neq B$				$3 \neq 4$, да
9	$X \leq (A+B)/2$				$3 \leq 3,5$, да
10	$B := \text{ЦЕЛ}((A+B)/2)$			3	$B := 3$
11	$A \neq B$				$3 \neq 3$, нет
12	Вывод А				Ответ: 3

Трассировочная таблица является моделью работы процессора при выполнении программы. Программа выполняется по шагам (первый столбец таблицы). В столбце «Команда алгоритма» отображается содержимое *регистра команд процессора*, куда помещается очередная команда. В столбце «Переменные» отображается содержимое *ячеек памяти* компьютера (или регистров памяти процессора), отведенных под переменные величины. В графе «Выполняемое действие» отражаются действия, выполняемые *арифметико-логическим устройством* процессора.

Таким образом, алгоритм в совокупности с трассировочной таблицей полностью моделируют процесс обработки информации, происходящий в компьютере.

Задания для самостоятельного выполнения.

Задание 1.

Способ вычисления наибольшего общего делителя НОД двух натуральных чисел (Алгоритм Евклида) в словесной форме можно описать так:

1. Если числа не равны, то большее из них заменить на разность большего и меньшего из чисел.
2. Если два числа равны, то за НОД принять любое из них, иначе перейти к выполнению пункта 1.

Данный алгоритм ориентирован на исполнителя-человека.

Задания:

- 1) Представьте данный алгоритм в виде блок-схемы.
- 2) Проведите трассировку алгоритма Евклида для нахождения НОД чисел 128 и 56.

Задание 2.

Ответьте на вопросы:

- 1) Почему алгоритм можно назвать моделью деятельности?
- 2) Почему алгоритм является информационной моделью?
- 3) Почему компьютер называют формальным исполнителем?
- 4) Что моделирует трассировочная таблица?