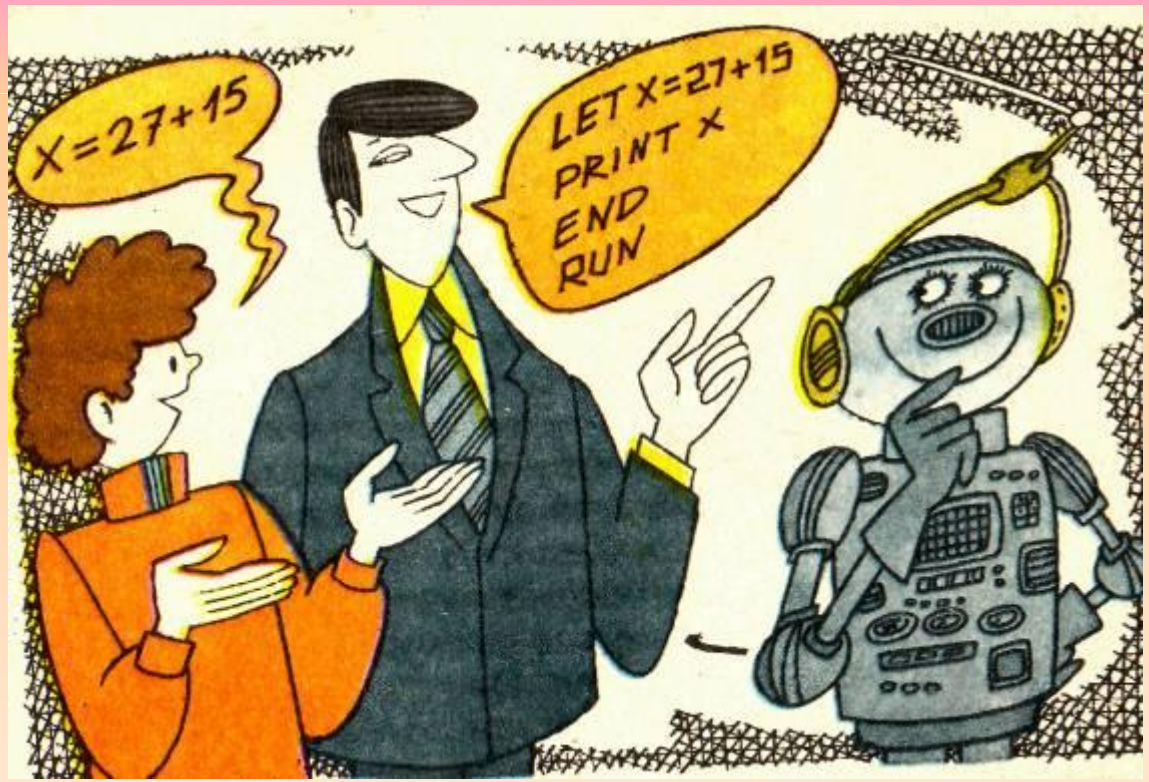
Первые шаги в программировании на языке Бейсик

Автор: Аверкина Т.П., учитель
МОУ «Тархановская СОШ»
Ичалковского района РМ

Один из первых, простых и удобных языков высокого уровня был разработан американскими учеными Дж. Кемени и Т. Куртцом. Назвали они его БЕЙСИК (BASIC) — Beginner's All-purpose Symbolic Instruction Code, что может быть переведено примерно как «многоцелевой символический код-инструкция для начинающих». Этот язык, усовершенствованный и дополненный, и теперь успешно используется в практике общения со многими компьютерами.

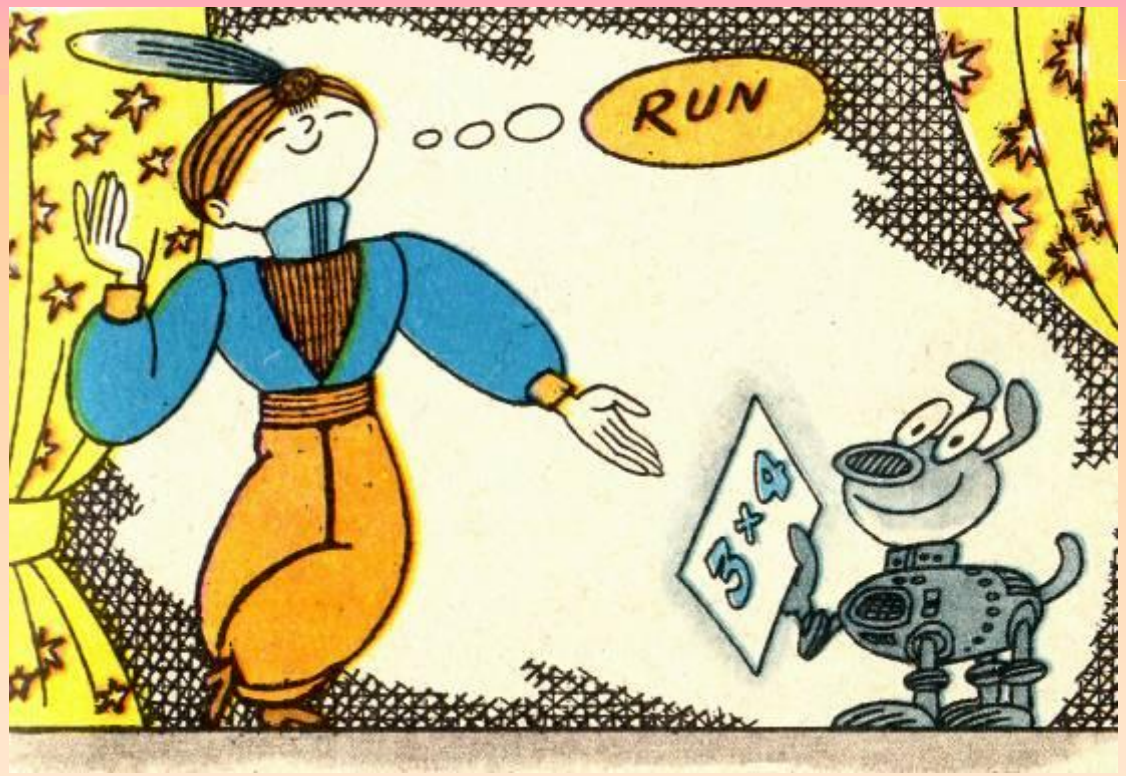
Пусть, например, вы хотите поручить компьютеру вычислить, сумму двух чисел: X=27+15.



Программа для вычисления X на языке БЕЙСИК записывается так:

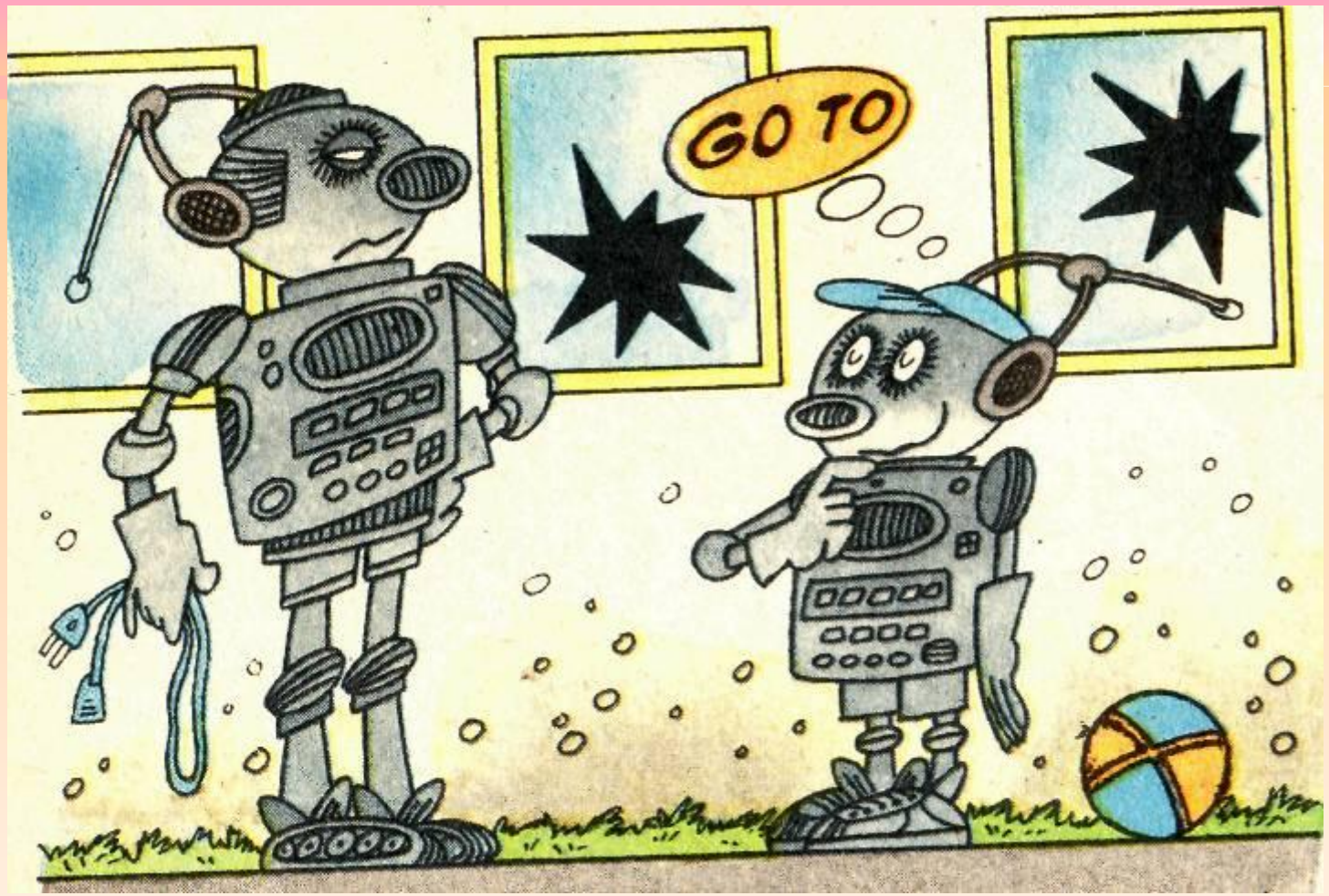
```
1  LET  X=27+15
2  PRINT X
3  END
```

По команде RUN машина проверяет, нет ли в программе какой-либо ошибки, и если нет, то добросовестно выполняет задание и печатает или высвечивает на экране дисплея результат — 42.



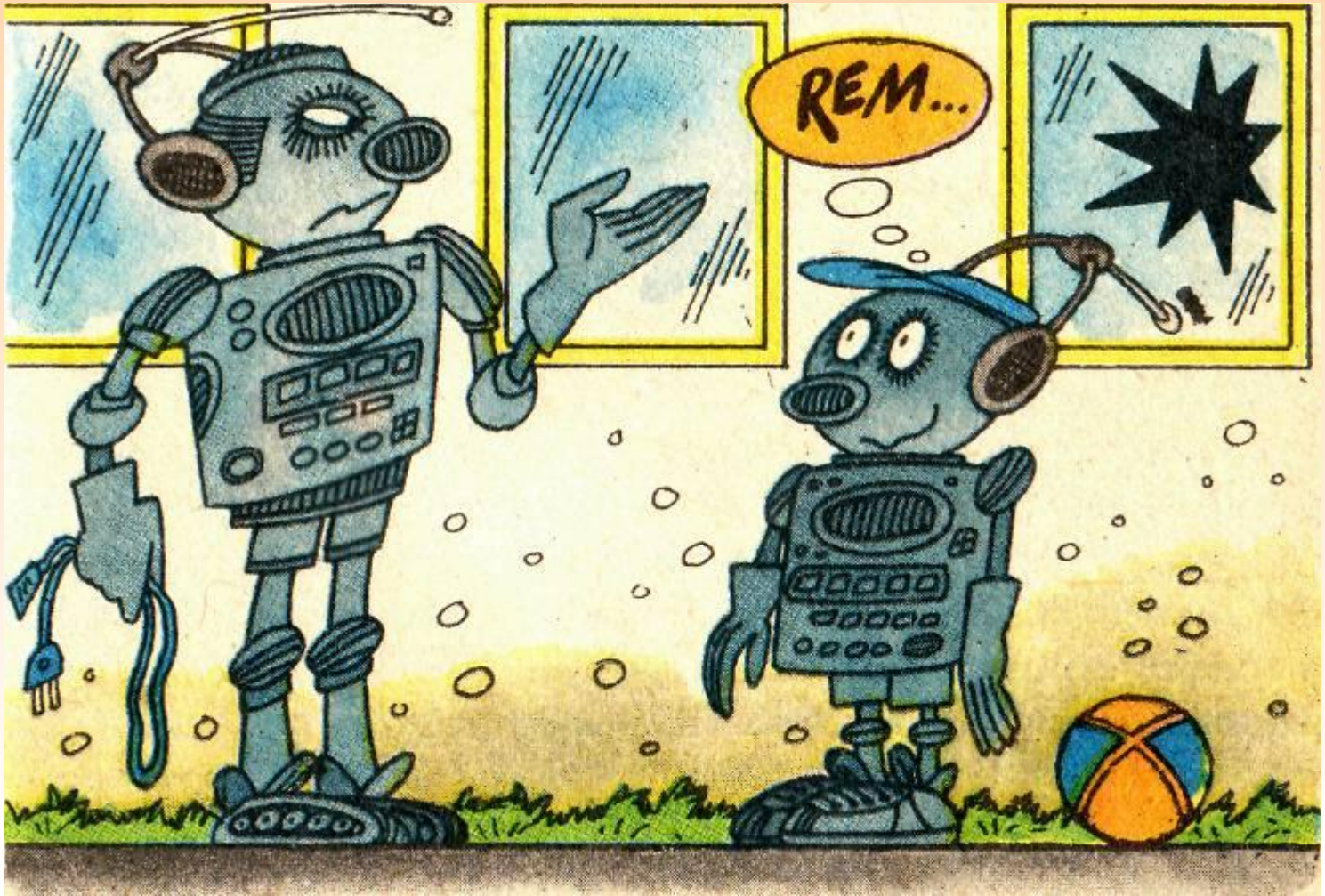
Для тех, кто не владеет английским языком, поясним слова, содержащиеся в командах (их называют операторами):

I.ET означает ПУСТЬ,
PRINT означает ПЕЧАТАТЬ,
END означает КОНЕЦ,
RUN означает ВЫПОЛНИТЬ.



Для того чтобы машина выполнила некоторое действие лишь при определенном условии, подается команда IF ...THEN... (т. е. ЕСЛИ ..., ТО ...). В случае необходимости изменить порядок выполнения действий подается команда GO TO ... (т. е. ПЕРЕЙТИ К ...).

Укажем еще оператор REM. **REM — это сокращение английского слова REMARK — разъяснение или замечание.** Команда, содержащая этот оператор, не выполняется машиной, она бывает полезна составителю программы.



Арифметические действия в командах на языке БЕЙСИК обозначаются символами:

+ сложение,

- вычитание,

*** умножение,**

/ деление.

Попробуем в виде примера составить программу для решения какой-нибудь математической задачи, простой, но достаточно громоздкой и утомительной для ее решения «вручную».

Вспомним старинную легенду об изобретателе шахмат индусском мудреце Сете. В ней рассказывается о том, как царь Ширам, познакомившийся с игрой в шахматы, был так восхищен ее остроумием и разнообразием возможных положений, что предложил мудрецу-изобретателю назначить себе награду за столь удачную выдумку.

И Сета пожелал получить награду пшеницей: за первую клетку шахматной доски — одно зернышко, за вторую — два, за третью — четыре и так далее, за каждую следующую клетку — вдвое против предыдущей.

Царю такое пожелание показалось слишком скромным.

Однако придворные математики, трудившиеся над вычислениями весь день и всю ночь, подсчитали, что во всех царских амбарах нет такого количества пшеницы, которое запросил хитрый Сета в награду за свое изобретение. Чтобы собрать столько зерна, пришлось бы засеять всю поверхность Земли, превратить в пахотные поля даже моря и океаны.

Воспользовавшись уже знакомыми нам операторами, запишем на языке БЕЙСИК программу подсчета количества пшеничных зерен, которые должен был выдать мудрецу Сете царь Ширам.

```
01  REM Награда изобретателю шахмат Сете
02  LET A=l
03  LET B=l
04  LET C=0
05  LET A=2*B+A
06  LET B=2*B
07  LET C=C+1
08  IF C<63 THEN 05
09  PRINT A «пшеничных зерен»
10  END
```

Если задать эту программу компьютеру, «понимающему» БЕЙСИК, то, выполнив задание, он напечатает или высветит на экране дисплея ответ: 18 446 144 073 709 551 615 пшеничных зерен.

1.8 446 144 073 709 551 615 E19

ПЕРВАЯ НАСТОЯЩАЯ ПРОГРАММА, УДОБНАЯ, СПОСОБНАЯ К ДИАЛОГУ...

Напишем теперь программу, которая будет вычислять расход бензина на 100 км пути.
Потребуем от нее, чтобы она:
— всякий раз поясняла, что ожидается от оператора;
— комментировала собственные действия;
— воспринимала пары данных (в нашем примере это количество пройденных километров и израсходованных литров бензина) и рассчитывала результат, учитывая все имеющиеся данные.

```
 10 LET  L=0
15  LET  K=0

 20 PRINT «ВВЕДИТЕ КМ И ЛИТРЫ»
 25 NPUT  K1, L1

 30 LET K=K + K1
 35 LET L=L+L1

 40 PRINT «ВВЕДИТЕ СЛОВО «ДА», ЕСЛИ ЕСТЬ
    ЕЩЕ ДАННЫЕ»
 45 INPUT F$
 47 IF F$=«ДА» GOTO 20

 50 LET P=L/K* 100
 55 PRINT «РАСХОД НА 100 КМ»; P

 60 STOP
 70 END
```

ДАВАЙТЕ ПОИГРАЕМ В «УГАДАЙКУ»!

Попробуем сыграть с компьютером в какую-нибудь несложную игру. Собственно говоря, играть мы будем не с самим компьютером, а с программой, управляющей им. Делал эту программу какой-то человек, программист. Иными словами, мы будем играть как бы с другим человеком, но только через посредничество компьютера.

Будем отгадывать числа, загаданные машиной. Каждому играющему дается восемь попыток для отгадывания. Программа построена так, чтобы каждое действие человека сопровождалось комментарием на экране. Удастся отгадать число — компьютер поздравит вас, а не удастся — утешит. Под конец игры он вежливо спросит, не желаете ли вы сыграть еще раз. Если вы согласны, он загадает новое число—и вновь у вас восемь попыток угадать его. Программа игры интересна большим числом разветвлений, а также наличием комментария. Это называют техникой диалога.

КОМПЬЮТЕР «БРОСАЕТ КОСТЬ»

Всем известно, что такое игральная кость. Это кубик, на каждой из сторон которого — цифры, от 1 до 6. Бросишь кубик — выпадет какое-то из этих чисел. Какое именно, зависит от случайности. Вероятность выпадания числа одинакова для всех шести граней кубика.

Можно написать программу, в которой цифры от 0 до 9 выпадают по случайному закону.

СЛУЧАЙНАЯ ФУНКЦИЯ

Во многих компьютерах имеется случайная функция, которая на языке БЕЙСИК обозначается как RND. Она всякий раз создает иное, случайным образом выбранное число между 0 и 1 — например, 0,382934802. Из него можно получить любое целое число между 0 и 9, умножив его на 10 и откинув все цифры после запятой.

Это достигается функцией INT, имеющейся в языке БЕЙСИК. Команда

N=INT(RND*10)

создает случайные числа так же, как это получается при бросании игральной кости. Если умножать случайно выпавшее число не на 10, а на 100, будут возникать целые числа между 00 и-99.

ПРОГРАММА «УГАДАЙКА»

```
10    PRINT «УГАДАЙ, КАКОЕ Я ЗАДУМАЛ ЧИСЛО МЕЖДУ 1 И 100»
20    PRINT «КСТАТИ, У ТЕБЯ ВОСЕМЬ ПОПЫТОК»
30     LET T=0
40     LET N=INT(RND » 100)+1
50     IF T=8 GOTO 180
60     LET T=T+1
70     PRINT «ТЕПЕРЬ ОТГАДЫВАЙ»
80     INPUT G
90      IF G=N GOTO 150
100    IF G>N GOTO 130
110    PRINT G; «МАЛО»
120    GOTO 50
130    PRINT G; «ВЕЛИКО»
140     GOTO 50
150   PRINT «ПОЗДРАВЛЯЮ, ТЫ ПОБЕДИЛ»
160   PRINT «А ВСЕ ЖЕ ТЫ ГАДАЛ»; T; «РАЗ»
170    GOTO 190
180 PRINT «ОЧЕНЬ ЖАЛЬ, НО ВЕДЬ У ТЕБЯ БЫЛО 8 ВОЗМОЖНОСТЕЙ»
185 PRINT «КСТАТИ, Я ЗАДУМАЛ ЧИСЛО»  N
190 PRINT «ДАВАЙ ЕЩЕ РАЗ СЫГРАЕМ? 1=ДА, 0=НЕТ»
200    INPUT B
210    IF B=0 GOTO 230
220    GOTO 30
230    PRINT «БОИШЬСЯ, ЧТО ОПЯТЬ ПРОИГРАЕШЬ?»
240    STOP
250     END
```

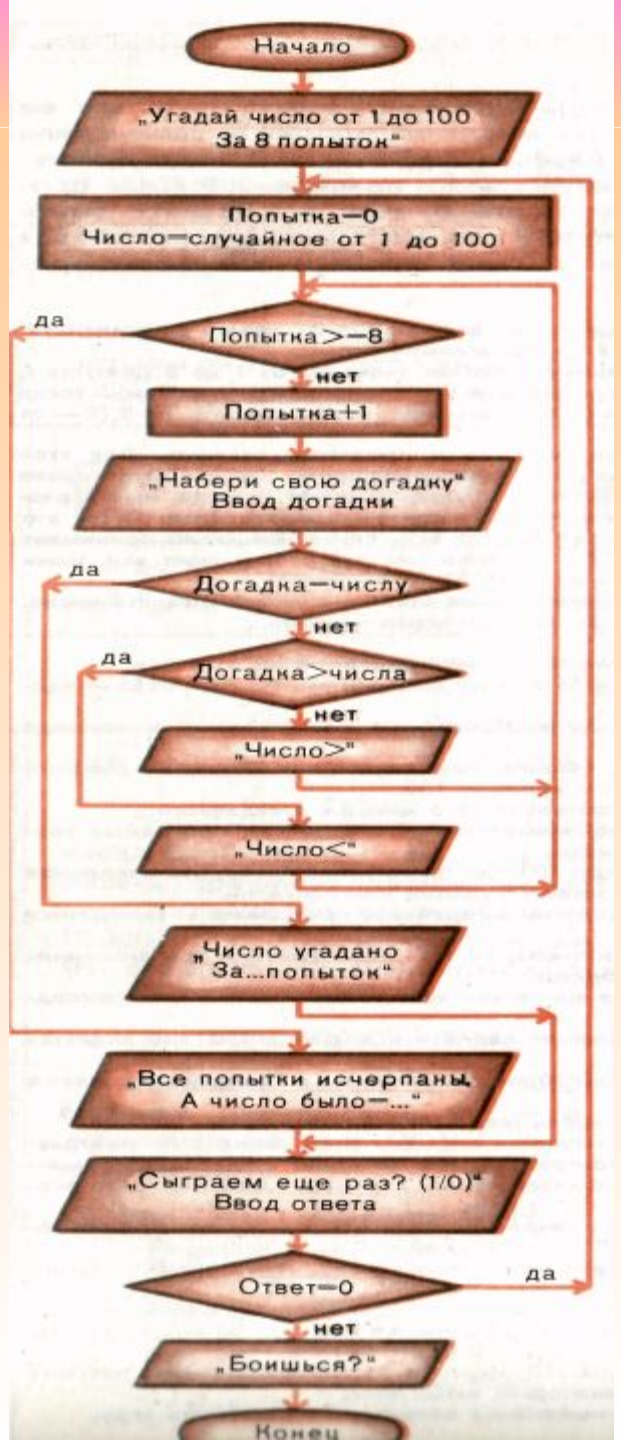
Эту же программу можно представить еще и графически — в виде условной «картинки», отображающей существующие, заложенные программистом связи.

Каждый шаг программы отображается в виде отдельного четырехугольника. Линии со стрелками показывают, что следует за каждым элементарным шагом, когда решается конкретная, сведенная до уровня «Да» — «Нет» задача.

Графическое «дерево», которое получается как результат такого представления программы, способно помочь ликвидировать многие ошибки, возникающие при записи программы. А ведь ошибки обязательно появляются...

Но и устранить их тоже нужно обязательно: иначе программа не будет работать или же начнет делать не то, что от нее ждут. «Картинка» помогает проследить правильность имеющихся связей между отдельными частями программы, их логику.

Для примера мы приведем программное «дерево» программы «Угадайка».



Литература

1. ЭВТ: Знакомимся, делаем, играем!/ Б. Игошев, М. Галагузова, Д.Комский. М.: Мол.гвардия, 1989.
2. Лоберг Р., Лутц Т. Домашний компьютер. М., Дет. лит., 1990.