

Информатика

Златопольский Дмитрий Михайлович

Кандидат технических наук, доцент кафедры информатики и прикладной математики Московского городского педагогического университета.



Компьютерный фокус «Отгадай число»

Хотите стать фокусником (☺) и удивить своих товарищей демонстрацией компьютерного фокуса? Если да, то эта статья – для вас.

В основе фокуса лежит достаточно известная игра с таким же названием. Напомним её правила. Один из участников игры загадывает целое число, например, меньшее 100. Второй участник должен отгадать это число, делая несколько попыток. В случае несовпадения названного и задуманного чисел первый участник сообщает, какое из них больше, после чего делается следующая попытка и т. д. до отгадывания. Вы, конечно, уже догадались, как должен действовать второй участник, – в первой попытке он должен назвать число 50, потом, в зависимости от ответа, – 25 или 75 и т. п. Такой принцип, позволяющий отгадать число за минимально возможное количество попыток, называется методом бинарного поиска (деления интервала поиска пополам).

Компьютерный фокус состоит в следующем. Фокусник – вы, сидящий у компьютера, – просит зрителей задумать любое целое число от 1 до 99 и сообщить его вам. Компью-

тер должен это число отгадать, выдавая на экран свои предположения. Зрители, глядя на эти предположения, называют любые слова, например, свои имена, которые вы вводите в компьютер с клавиатуры. Компьютер отгадывает число, причём делает это методом бинарного поиска, т. е. за минимально возможное количество попыток. Ясно, что при демонстрации фокуса зрители не сообщают о том, больше или меньше задуманное ими число, но, тем не менее, компьютер его отгадает.



Секрет фокуса: вы, набирая на компьютере очередное названное слово (например, чье-то имя), незаметно для зрителей вводите в компьютер информацию о том, справа или слева в числовом ряду находится задуманное число от последнего предположения машины.

Передать в программу эту информацию можно различными способами. В следующей программе на языке Паскаль это делается с помощью пробела, который вводится или не вводится в конце слова. Ос-

новные величины, используемые в программе:

min – левая граница интервала, в котором расположено искомое число;

max – то же, правая граница;

sered – середина этого интервала;

nomer_popitki – номер попытки;

name – имя одного из зрителей;

otgadal – имеет значение «у» (латинская буква) или «д», если число угадано, а иначе – «n» или «н».

Итак, программа:

```
Program Fokus;  
Uses CRT;  
Var min, max, sered, nomer_popitki: byte;  
    name: string[20]; otgadal: char;  
BEGIN  
  clrscr;  
  writeln('Задумайте целое число в интервале от 1 до 99');  
  writeln('Сейчас я отгадаю его за несколько попыток!');  
  {Приостановка программы до нажатия любой клавиши}  
  Repeat  
  until KeyPressed;  
  min := 1; max := 99;  
  nomer_popitki := 0;  
  repeat {Цикл "угадывания" числа}  
  begin  
    {Очередная попытка}  
    nomer_popitki := nomer_popitki + 1;  
    sered := (min + max) div 2;  
    clrscr;  
    writeln(nomer_popitki, '-я попытка. Вы задумали число ',  
            sered);  
    writeln('Я прав? (y/n или д/н)');  
    {Вы задаете аналогичный вопрос зрителям и вводите ответ}  
    repeat  
      otgadal := ReadKey {Ждем нажатия клавиш "y", "n", "д"  
        или "н"}  
    until (otgadal = 'y') or (otgadal = 'n')  
      or (otgadal = 'д') or (otgadal = 'н');  
    if (otgadal = 'n') or (otgadal = 'н') then writeln('НЕТ!')  
    else writeln('ДА!');  
    {Приостановка программы до нажатия любой клавиши}  
  repeat  
  until KeyPressed;  
  {Если ответ неверный}  
  if (otgadal = 'n') or (otgadal = 'н') then
```

```
begin
  writeln('Кто это сказал? Как Вас зовут?');
  {Вводим имя зрителя}
  readln(name);
  {Меняем границы интервала поиска}
  if name[Length(name)] = ' '
  {Если последний символ имени - пробел}
  then min := sered
  else max := sered;
  end
end
until (otgadal = 'y') or (otgadal = 'д');
clrscr;
writeln('Число отгадано!');
writeln('Количество попыток: ', nomer_popitki);
writeln('Какой я умный!');
{Приостановка программы до нажатия любой клавиши}
repeat
until KeyPressed;
END.
```

Примечание. Чтобы при определении задуманного числа увеличить значение *min*, в конец имени зрителя необходимо ввести пробел, для уменьшения значения *max* он не нужен.

Однако не исключено, что при таком способе передачи информации в программу вас могут «раскусить» (почти «расфокусить» – ☺), уличив в «подсказке». Более эффективным является другой способ. Этот способ заключается в выдерживании различных пауз между вводом, например, первого и второго символов имени одного из зрителей.

Для ввода имени при этом следует использовать средства, контролирующие нажатие одной клавиши, а также функции и процедуры, отслеживающие время.

Идея такая – когда нажат первый символ имени, запомнить время нажатия, когда второй – также сделать это и определить промежуток времени между двумя значениями.

В представленной ниже программе на языке Паскаль используются следующие величины (кроме

величин *min*, *max*, *sered*, *otgadal* и *nomer_popitki*, применявшихся в первом варианте программы):

s – очередной символ имени;

i – порядковый номер символа в имени зрителя;

time1 – время в момент ввода первого символа (если быть точным, то эта величина представляет собой



общее число сотых долей секунды, прошедших с начала суток, – см. ниже о процедуре *GetTime*);

time2 – то же для второго (уточнение – аналогичное);

time – промежуток времени между вводом первого и второго символов;

predel – предельное значение промежутка времени между вводом первого и второго символов, в зависимости от которого происходит изменение границ интервала поиска *min* и *max*. Это значение подбирается опытным путём.

Передачу в программу информации, в зависимости от которой происходит изменение значений *min* и *max*, можно проводить по следующему правилу: если *time* > *predel*, то задуманное число больше последнего предположения компьютера (то есть должно быть увеличено значение *min*), в противном случае – меньше (должно быть уменьшено значение *max*).

Теперь о том, как рассчитать значения *time1* и *time2*. Для этого следует применить процедуру *GetTime*. Она возвращает 4 значения – её аргументы:

1) количество полных часов, прошедших с начала суток (имя величины – *ch*);

2) количество полных минут, прошедших с начала очередного часа (*m*);

3) количество полных секунд, прошедших с начала очередной минуты (*s*);

4) количество сотых долей секунды, прошедших с начала очередной секунды (*s100*).

Зная эти значения, можем считать значения *time1* и *time2* следующим образом:

$$ch \cdot 24 \cdot 3600 \cdot 100 + m \cdot 60 \cdot 100 + s \cdot 100 + sec100.$$

Для использования процедуры *GetTime* в программе необходимо в разделе описаний, в подразделе *Uses* указать модуль *Dos*.

Ещё два комментария к представленной ниже программе, с помощью которой демонстрируется фокус:

1) функция *ReadKey* символьного типа возвращает символ, соответствующий нажатой клавише (см. первый вариант программы);

2) тот факт, что ввод имени закончился и нажата клавиша **Enter** (код которой равен 13), фиксируется по условию *c = #13*.



Вся программа имеет вид:

```
Uses CRT, DOS;
Var min, max, sered, nomer_popitki: byte;
    otgadal, c: char;
    ch, m, sec, sec100: word;
    time1, time2, time: longint;
    predel, i: integer;
BEGIN
  clrscr;
  writeln('Задумайте целое число в интервале от 1 до 99');
  writeln('Сейчас я отгадаю его за несколько попыток!');
```

```
{Приостановка программы до нажатия любой клавиши}
repeat
until KeyPressed;
min := 1; max := 99;
predel := 200; {Значение подбирается опытным путём}
nomer_popitki := 0;
repeat {Цикл "угадывания" числа}
begin
    {Очередная попытка}
    clrscr;
    nomer_popitki := nomer_popitki + 1;
    sered := (min + max) div 2;
    writeln(nomer_popitki, '-я попытка. Вы задумали число ',
            sered);
    writeln('Я прав? (y/n или д/н)');
repeat
    otgadal := ReadKey {Ждем нажатия клавиш "y", "n",
    "д" или "н"}
until (otgadal = 'y') or (otgadal = 'n') or (otgadal = 'д')
    or (otgadal = 'н');
if (otgadal = 'n') or (otgadal = 'н')
    then writeln('НЕТ!')
    else writeln('ДА!');
{Приостановка программы до нажатия любой клавиши}
repeat
until KeyPressed;
{Если ответ неверный}
if (otgadal = 'n') or (otgadal = 'н') then
begin
    writeln('Кто это сказал? Как Вас зовут?');
    i := 0;
    repeat {Цикл ввода символов имени}
    i := i + 1; {Номер очередного символа}
    c := ReadKey; {Ждем нажатия клавиши}
    if c <> #13 {Если ввод символов имени не окончен}
    then {Повторяем символ на экране}
        write(c);
    case i of
        {Нас интересуют только первый}
        1: begin
            GetTime(ch, m, sec, sec100);
            time1 := ch * 24 * 3600 * 100 + m * 60 * 100
                + sec * 100 + sec100
        end;
        {и второй символ имени}
        2: begin
            GetTime(ch, m, sec, sec100);
            time2 := ch * 24 * 3600 * 100 + m * 60 * 100
                + sec * 100 + sec100;
            time := time2 - time1
```

```
        end
    end {case}
    until c = #13; {До нажатия клавиши Enter
                    после ввода имени}
    {Меняем границы интервала поиска}
    if time > predel then min := sered
        else max := sered
    end
end
until (otgadad = 'y') or (otgadad = 'д');
clrscr;
writeln('Число отгадано!');
writeln('Количество попыток - ', nomer_popytki);
writeln('Какой я умный!');
{Приостановка программы до нажатия любой клавиши}
repeat
until KeyPressed
END.
```

В заключение ещё раз обратим внимание на тот факт, что главный эффект фокуса в том, что компьютер отгадывает число (с вашей по-

мощью), якобы «не зная», какое из чисел – задуманное зрителями или его предположение – больше.

Задание для самостоятельной работы

Разработайте (на языке программирования, который вы изучаете) вариант программы для демонстрации фокуса, в котором вводится не имя одного из зрителей, а только слова «ДА» и

«НЕТ», при этом при вводе слова «НЕТ» и происходит передача в программу необходимой информации (любым из двух описанных в статье способов).

Успехов в демонстрации фокуса!

Юмор Юмор Юмор Юмор Юмор Юмор

Директор компьютерной фирмы вызывает подчинённого и сообщает:
– В ходе вашей работы в нашей компании произошла ошибка. Попробуйте уволиться и устроиться ещё раз.

- Нажмите любую клавишу.
- Эту можно?
- Нет.

С появлением Интернета вопрос «Как убить время?» не актуален. Актуален вопрос «Как выспаться за 2 часа?».

99% ошибок компьютера сидит в полуметре от монитора.