Rozhledy matematicko-fyzikální

Stanislav Trávníček

Malá násobilka textově

Rozhledy matematicko-fyzikální, Vol. 89 (2014), No. 4, 27–31

Persistent URL: http://dml.cz/dmlcz/146597

Malá násobilka textově

Stanislav Trávníček, PřF UP, Olomouc

Abstract. The article shows a simple case of a text analysis and deciphering on the example of multiplication. The sample program makes multiplications in a text mode.

Začínající programátor – například amatér, který se chce naučit programovat jen pro svou potřebu a pro své potěšení – může získávat cenné zkušenosti i při řešení relativně jednoduchých problémů, zejména takových, kde se nabízí více cest, jak se dopracovat výsledku. O jednom takovém problému nyní pojednáme.

Chceme vytvořit počítačový program na malou násobilku, tj. od 0×0 po 10×10, ale takový, že příklady se zadávají textem, jako textový řetězec. Např. místo 4×9 zapíšeme *čtyři krát devět*. Je zřejmé, že tady nepůjde o matematiku, ale výhradně o práci s texty a o zvládnutí ne zcela jednoduché situace. Zaměříme se na co možno zcela jednoduchý program. Připomeňme zde jiný program na malou násobilku [1], kde však nešlo o hru, ale o pomůcku na procvičování malé násobilky.

Nejprve je třeba si stanovit, jak zapisovat zadání (to už jsme si vlastně stanovili v předchozím odstavci), a jak přesně musí být zadání napsáno. Nebuďme však příliš přísní a dopusťme, že předchozí příklad může být zadán i jinak, např.: *čtyři krát devět, čtyřikrátdevět, ctyri krat devet, ctyrikrat devět, ctyři kratdevět.* Tato blahovůle se samozřejmě odrazí na určitých komplikacích v programové analýze. Načtení výrazu ovšem složité není, označme jej *Expr*. Řekli jsme si, že chceme program co nejjednodušší, a budeme tedy předpokládat, že uživatel nebude úmyslně zadávat chybné vstupy, a proto budeme zachycovat jen hrubé chyby.

Analýzu textu, tedy zjištění, která čísla text obsahuje, svěříme funkci *Zjisti*, která nám po analýze poskytne číslo 0 až 10, nebo sdělí, že došlo k chybě. Daným výrazem budeme postupovat pomocí indexu *Ind*, který má na počátku nastavenu hodnotu 1. Pokud se však na prvním nebo i dalších místech výrazu vyskytuje mezera, zvětší se index až k prvnímu písmenu zadaného čísla. (Tedy mezery v jakémkoli počtu na počátku výrazu, mezi slovy ani na konci výrazu nevadí.) Ukazuje se, že v některých

případech stačí už toto písmeno ke stanovení, o jaké číslo jde. Jsou to písmena: n = nula, j = jedna, t = tři, c = čtyři, p = pět, o = osm. Číslo čtyři však můžeme dostat i z počátečního č, ale v programu nepoužijeme kódování češtiny a půjdeme na to jinak. Zůstávají nám dále: d = dvě, devět, deset, s = šest a sedm a č, š.

Začněme případem 's'. Druhý znak je v obou případech 'e', takže musíme sáhnout až ke znaku třetímu, kdy 'd' nám dá 7 a 's' dává 6. Písmeno 'd' dává tři případy. Je-li druhé písmeno 'v', pak jde o číslo 2, ale devět a deset je třeba rozlišit až třetím písmenem, 'v' dává 9, 's' pak značí 10. Konečně máme případy počátečního 'č' a 'š', které zahrneme do případu else. Zde jsme mimořádně testovali současně druhý a třetí znak (ale stačil druhý), takže 'ty' nám dalo 4 a 'es' pak 6.

Druhý činitel součinu následuje po slovu 'krát', takže se ptáme na polohu 'kr' v daném výrazu, index nastavíme za slovo krát a funkce *Zjisti* najde druhého činitele.

Po získání výsledku násobení přichází druhá úloha – textový výstup výsledku. Případy výsledku 100 a 0 jsou mimořádné a jsou vyřešeny zvlášť. Pak se pracuje se dvěma textovými řadami, čísel menších než 20 a ostatních. V prvním případě žádné komplikace nenastanou, ve druhém případě se tisknou nejprve desítky ('dvacet', 'třicet', ...) a v případě, že jednotky nejsou 0, dotiskují se jednotky. Místo tisku je vypočteno, aby celkový dojem zápisu byl příjemný, včetně tečky za výsledkem.

Program je postaven jako smyčka, takže po jeho spuštění lze provádět více výpočtů. Je samozřejmé, že tím, že jsme obešli redundanci (teoretickou nadbytečnost) zápisu čísel, můžeme někdy dostat nějaký nesmyslný výsledek na základě nesmyslného zadání, ale vraťme se k úmluvě, že uživatel toto provádět nechce. Na obr. 1 je uveden obsah obrazovky po provedeném výpočtu 4×9.

```
Malá násobilka textově

Zadejte příklad jako text, výsledek dostanete opět jako text.

Zadávat lze česky: čtyřikrát šest nebo čtyři krát šest nebo
ctyrikrat sest, nebo ctyrikratsest, apod. Mezery před, mezi
nebo za slovy nevadí.

Vypočítej: čtyřikrát devět je třicet šest.

Další úloha? (A/N)_
```

Obr. 1

Výpis programu:

```
program MNtext;
uses Crt;

const
  Num1: array [0..18] of String =('','jedna','dvě','tři', 'čtyři',
'pět','šest','sedm','osm','devět','deset','', 'dvanáct',' ',
'čtrnáct','patnáct','šestnáct','','osmnáct');
  Num2: array [1..9] of String =('','dvacet','třicet', 'čtyřicet',
'padesát','šedesát','sedmdesát','osmdesát', 'devadesát');
  BNad = yellow+16*red;
  BVst = white;
  BVys = yellow;
  BInf = lightgreen;

var
  Ind, M, N, S: Byte;
  Expr: string;
  Err: Boolean;
  Dale: Char;

function Zjisti: Integer;
var Z: Integer;
begin
  while Expr[Ind] = ‘‘do Inc(Ind);
  case Expr[Ind] of
  'n': Z := 0;
  'j': Z := 1;
  't': Z := 3;
  'c': Z := 4;
  'p': Z := 5;
  'o': Z := 8;
  's': if Expr[Ind+2] = 'd' then Z := 7
    else
      if Expr[Ind+2] = 's' then Z := 6
      else Err := true;
  'd': if Expr[Ind+1] = 'v' then Z := 2
    else
      if Expr[Ind+2] = 's' then Z := 10
```

```
      else
        if Expr[Ind+2] = 'v' then Z := 9
        else Err := true;
  else
    if (Expr[Ind+1]='t') and (Expr[Ind+2]='y') then Z := 4
    else
      if (Expr[Ind+1]='e') and (Expr[Ind+2]='s') then Z := 6
      else Err := true
  end;
  Zjisti := Z
end; {Zjisti}

procedure Tiskni(X: Integer);
begin
  Write('je ');
  case X of
  100: Write('sto');
  0: Write('nula');
  else
    if X < 20 then Write(Num1[X])
    else
    begin
      Write(Num2[X div 10]);
      if (X mod 10) <> 0 then Write(' ',Num1[X mod 10])
    end
  end;
  Write('.') end; {Tiskni}

begin {program}
  repeat
    ClrScr;
    GotoXY(2,2);
    TextAttr := BNad;
    Write('Malá násobilka textově');
    GotoXY(2,4);
    TextAttr := BInf;
    Write('Zadejte příklad jako text,');
    WriteLn(' výsledek dostanete opět jako text.');
    WriteLn;
```

```
    Write(' Zadávat lze česky: čtyřikrát šest nebo ');
    WriteLn('čtyři krát šest nebo');
    Write(' ctyrikrat sest, nebo ctarikratsest, apod.');
    WriteLn(' Mezery před, mezi');
    WruteLn(‘nebo za slovy nevadí.‘);
    Err := false;
    repeat
      Ind := 1;
      GotoXY(2,11);
      Write(' ');
      GotoXY(2,11);
      TextAttr := BVst;
      Write('Vypočítej: ');
      ReadLn(Expr);
      M := Zjisti;
      if not Err then Ind := Pos('kr',Expr) + 4;
      N := Zjisti
    until Err = false;
    S := M * N;
    TextAttr := BVys;
    GotoXY(Length(Expr)+14,11);
    Tiskni(S);
    GotoXY(2,14);
    TextAttr := BVst;
    Write('Další úloha? (A/N)');
    Dale := ReadKey;
  until (Dale <> 'A') and (Dale <> 'a');
  WriteLn; TextAttr := BInf;
  Write(' Konec.');
  ReadLn
end. {program}
```

Je samozřejmé, že se s tímto námětem dá ještě dále pracovat, například zdokonalit kontrolu vstupního výrazu, nebo např. sestavit program, který akceptuje zadání slovy i čísly.

Literatura

[1] Trávníček, S.: Násobík. *MFI* **23**, č. 5 (2014), s. 390–394.