# INFORMATIKA

# Počítačová bezpečnost ve výuce informatiky

## (7. část: snadné transpoziční šifry a výuka programování)

*MICHAL MUSÍLEK – ŠTĚPÁN HUBÁLOVSKÝ*

Přírodovědecká fakulta Univerzity Hradec Králové

V minulých článcích našeho seriálu jsme se seznámili s pojmy otevřený text, šifrový text, šifrování, dešifrování, luštění a frekvenční analýza textu. Ve 3. části jsme také zmínili možný výsledek frekvenční analýzy, kdy v analyzovaném v šifrovém textu je nejčetnějším písmenem E, dalšími A, I, O, N, atd., stejně jako v běžném českém otevřeném textu. Potom se pravděpodobně jedná o text zašifrovaný změnou pořadí písmen tvořících zprávu, tedy transpozici. Přestože jsme ještě zdaleka nepopsali všechny existující typy substitučních šifer, podíváme se dnes pro změnu na snadné transpoziční šifry a na realizaci jejich algoritmů v programovacím jazyce snadno dostupném našim žákům.

Jednou z možností realizace algoritmu je vložení skriptu do HTML stránky. Skript je potom interpretován prostřednictvím webového prohlížeče, aniž bychom pro jeho běh potřebovali jakékoliv jiné prostředí. Jako skriptovací jazyk můžeme použít např. JavaScript, nebo VBScript. v našich ukázkách jsme se rozhodli pro JavaScript [1]. k editaci zdrojového kódu lze použít program „Poznámkový blok“ z příslušenství operačního systému Windows. Nejprve si připravíme HTML dokument, do kterého pak budeme vkládat různé skripty:

```
<html>
 <head>
  <title>Transpoziční šifry</title>
 </head>
 <script type="text/javascript" language="JavaScript">
 <!--
   /*Zde budou uvedeny konkrétní skripty pro šifrování*/
 //-->
 </script>
 <body>
  <h1>Transpoziční šifry</h1>
  <form name="texty">
    <table border="0">
      <tr><td>OT: </td><td><input type="text" name="ot"
                    maxlength="150" size="150"></td></tr>
      <tr><td>ŠT: </td><td><input type="text" name="st"
                    maxlength="150" size="150"></td></tr>
      <tr><td> </td><td>
  <!-- Následující řádky s tlačítky budou doplňovány: -->
  <input type="button" value="Odzadu" onClick="odzadu();">
    </td></tr>
   </table>
  </form>
 </body>
</html>
```

Nejjednodušší transpozice je založena na přepisu celého textu odzadu. Budeme-li pracovat s datovým typem „řetězec znaků“ a označíme-li OT řetězec představující otevřený text a ŠT řetězec představující šifrový text, můžeme algoritmus pro přepis celého textu odzadu rozepsat do následujících kroků:

1. Zjisti délku řetězce OT. Nastav hodnotu proměnné i na délku řetězce OT.
2. Nastav hodnotu ŠT na prázdný řetězec.
3. Přečti $i$-tý znak řetězce OT a připoj ho na konec řetězce ŠT.
4. Zmenši hodnotu $i$ o 1. Je-li $i > 0$, jdi na 3. krok, jinak skonči.

Převod algoritmu do JavaScriptu je ovlivněn číslováním pozic znaků v řetězci s délkou $n$, které není, jak bychom nejspíš čekali, od 1 do $n$, ale od 0 do $n-1$. V názvech proměnných také musíme vynechat diakritiku. Celý podprogram je uvozen klíčovým slovem function, protože jazyk JavaScript nerozlišuje z hlediska syntaxe mezi podprogramy, procedurami a funkcemi. Do připraveného HTML dokumentu vložíme skript:

```
function odzadu()
      {
        OT = document.texty.ot.value;
        n  = OT.length
        ST = '';
        for (i = n-1; i > -1; i--)
          {ST += OT.charAt(i)};
        document.texty.st.value = ST;
      }
```

Na konci tabulky formuláře využijeme tlačítko:

```
<input type="button" value="Odzadu" onClick="odzadu();">
```

Jinou, algoritmicky o něco složitější transpozicí, je psaní jednotlivých slov otevřeného textu odzadu. Začněme opět slovním popisem algoritmu:

1. Rozděl řetězec OT na podřetězce, oddělovačem, který určí toto rozdělení, je mezera.
2. Zjisti počet podřetězců.
3. Jednotlivé podřetězce přepiš odzadu dopředu s využitím dříve popsaného algoritmu.
4. Sestav ŠT z jednotlivých odzadu přepsaných podřetězců.

Kromě řetězců OT a ŠT budeme nyní muset pracovat také s polem podřetězců PP. Počet podřetězců označíme $k$, zatímco délka jednotlivého podřetězce bude i nadále $n$. Cykly budou mít řídící proměnné $j$ a $i$. Podprogram v JavaScriptu může vypadat např. takto:

```
function poSlovech()
      {
```

```
    OT = document.texty.ot.value;
    PP = OT.split(' '); /* rozděl na podřetězce */
    k  = PP.length;    /* zjisti počet podřetězců */
    ST = '';
    for (j = 0; j < k; j++)
      {
        n = PP[j].length;
        for (i = n-1; i~> -1; i--)
          {ST += PP[j].charAt(i)};
        ST += ' ';
      }
    document.texty.st.value = ST;
  }
```

Na konci tabulky formuláře vložíme další tlačítko:

```
<input type=
"button" value="Poslovech " onClick="poSlovech();">
```

Třetí možností přepisu odzadu je vytvoření pětimístných skupin znaků, oddělených mezerou, na něž potom uplatníme výše uvedený algoritmus namísto na slova. Příslušný skript necháme jako cvičení čtenářům. Přepis celého textu nebo jeho částí (slov, skupin) odzadu je samozřejmě jen jednou z mnoha snadných transpozičních šifer, vhodných jako úlohy pro výuku algoritmizace a programování.

**Šifrování „podle plotu“**

Další skupinou šifer, které dle *Simona Singha* [3] i *Pavla Vondrušky* [4] s oblibou používají pro psaní tajných zpráv školáci v Anglii, jsou šifry „podle plotu“ (angl. *Rail Fence Cipher*). Existují různé varianty těchto šifer, které se liší počtem řádků a popřípadě také tvarem cik-cak lomené čáry, která spojuje písmena otevřeného textu, zapsaná do tabulky představující plot. Úplně nejjednodušší je použití tabulky se dvěma řádky, do které zapisujeme otevřený text střídavě do prvního a druhého řádku, a to bez diakritiky, interpunkčních znamének a mezer. Výsledný šifrový text pak čteme po řádcích a přepíšeme jej do pětimístných skupin. Ukažme si šifrování na příkladu. Máme zašifrovat text: „Tak dlouho se chodí se džbánem pro vodu, až se ucho utrhne.“
Přepis do tabulky:

| T | | K | | L | | U | | O | | E | | H | | D | | S | | D | | B | | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | | D | | O | | H | | S | | C | | O | | I | | E | | Z | | A | |

| | M | | R | | V | | D | | A | | S | | U | | H | | U | | R | | N | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E | | P | | O | | O | | U | | Z | | E | | C | | O | | T | | H | | E |

Šifrový text:

```
TKLUO EHDSD BNMRV DASUH URNAD OHSCO IEZAE POOUZ ECOTH E
```

Vidíme, že výsledný text bychom v tomto případě získali také „rozpočítáním“ znaků zprávy na první a druhé (liché a sudé) a následným vypsáním nejprve všech prvých a potom všech druhých. Zkusme, nejprve opět slovně, popsat algoritmus šifrování:

I. Odstraň z textu všechnu diakritiku, interpunkční znaménka a mezery a zapiš jej pouze velkými písmeny (tj. malá písmena změň na velká).

II. Rozpočítej znaky textu na první a druhé a vytvoř dva podřetězce, ze kterých se potom složí výsledný šifrový text.

III. Vypiš výsledný šifrový text a přitom za každých pět znaků vlož mezeru (tj. vytvoř pětimístné skupiny znaků).

Tento stručný zápis algoritmu některým žákům stačí, aby se s využitím referenční příručky JavaScriptu [1] pustili do programování. Většina ale bude pravděpodobně potřebovat jemnější členění postupu do menších kroků:

1. $i = 1$; $n =$ délka OT; ŠT = prázdný řetězec.
2. Vezmi $i$-tý znak OT
   a. Je to písmeno bez diakritiky? Jestliže ANO, připoj ho k ŠT.
   b. Je to písmeno s diakritikou? Jestliže ANO, nahraď ho odpovídajícím znakem bez diakritiky a připoj ho k ŠT.
3. Zvětši $i$ o 1.
4. Je $i > n$? NE... jdi na 2. krok. ANO... pokračuj 5. krokem.

5. Přiřaď hodnotu ŠT do OT. Tím je hotova I. fáze algoritmu – viz hrubý popis.

6. $i = 1$; $n$ = délka upraveného OT; ŠT, PRVNÍ, DRUHÉ = prázdný řetězec.

7. Vezmi $i$-tý znak OT a. Je i liché číslo? Připoj $i$-tý znak OT k řetězci PRVNÍ. b. Je i sudé číslo? Připoj $i$-tý znak OT k řetězci DRUHÉ.

8. Zvětši $i$ o 1.

9. Je $i > n$? NE... jdi na 7. krok. ANO... pokračuj 10. krokem.

10. Proveď spojení řetězců ŠT = PRVNÍ + DRUHÉ. Tím je hotová II. fáze algoritmu.

11. $i = 1$; OT = ŠT; ŠT = prázdný řetězec.

12. Připoj $i$-tý znak OT k ŠT.

13. Je i dělitelné pěti a zároveň $i > 0$? Jestliže ANO, připoj k ŠT mezeru.

14. Je $i > n$? NE... jdi na 12. krok. ANO... pokračuj 15. krokem.

15. Vypiš ŠT. Konec.

Příslušný program v JavaScriptu je výhodné rozepsat do několika podprogramů, protože odstranění diakritiky, interpunkce a mezer se nám ještě může hodit i v jiných typech šifer, stejně jako rozdělení dlouhého řetězce na pětimístné skupiny oddělené mezerami. Skript, který spojí všechny tři fáze do jednoho procesu bude uveden jako poslední a bude vypadat následovně:

```
function podlePlotu()
  {
    vycisteni();
    rozpocitani();
    poPeti();
  }
```

Podprogram „vyčištění“, který odstraní diakritiku, interpunkci a mezery, lze v JavaScriptu zapsat následovně:

```
function vycisteni()
 {
   OT = document.texty.ot.value;/* vezmi otevřený text */
   OT = OT.toUpperCase();   /* převeď na velká písmena */
   n  = OT.length;          /* zjisti délku řetězce */
   ST = '';                 /* ŠT je zatím prázdný */
   for (i = 0; i~< n; i++)
    {
     znakOT = OT.charAt(i);
     znakST = '';    /* jiné znaky než písmena neber */
     if (('A' <= znakOT) && (znakOT <= 'Z'))
          {znakST = znakOT};
     if (znakOT == "Á")
         {znakST = 'A'};   /*odstraň čárku */
     if (znakOT == "Č")
         {znakST = 'C'};  /* odstraň háček */
  //... zde vyjmenovat všechna písmena s diakritikou!!
     if (znakOT == "Ž")
         {znakST = 'Z'};
     ST += znakST;    /* přidej znak k šifrovému textu */
    }
   document.texty.ot.value = ST;
       /*zapiš ŠT do horního pole*/
 }
```

Podprogram pro rozpočítání (rozdělení „podle plotu“) je sám o sobě velmi jednoduchý:

```
function rozpocitani()
  {
    OT = document.texty.ot.value;
    n = OT.length;
    PRVNI = '';
    DRUHE = '';
    for (i = 0; i~< n; i++)
      if (i % 2 == 0) {PRVNI += OT.charAt(i)}
      else            {DRUHE += OT.charAt(i)};
    ST = PRVNI + DRUHE;
    document.texty.st.value = ST;
```

```
  }
```

Poslední podprogram vloží za každé páté písmeno šifrového textu mezeru:

```
function poPeti()
  {
    OT = document.texty.st.value;
    n = OT.length;
    ST = '';
    for (i = 0; i~< n; i++)
      {
        ST += OT.charAt(i);
        if (i % 5 == 4) {ST += ' '}; /* vloží mezeru */
      }
    document.texty.st.value = ST;
  }
```

Nakonec přidáme tlačítko pro spuštění celého skriptu:

```
<input type=
"button" value="Podle plotu" onClick="podlePlotu();">
```

Kromě této nejjednodušší varianty můžeme šifrovat „podle plotu“ v tabulce se třemi, či více řádky a rozmístění písmen se může řídit o něco složitějším schématem než je klikatá čára od horní hranice k dolní a zpět. Ukažme si takové šifrování opět na příkladu:

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | | | | | | E | | E | | | | | | U | | I | | | | |
| | E | | D | | M | | S | | V | | A | | D | | B | | C | | C | |
| | | J | | E | | | | | | P | | R | | | | | | I | | H |

Otevřený text: Sejdeme se v Pardubicích.
Šifrový text: SEEUI EDMSV ADBCC JEPRI H

Všimněte si, že zatímco u šifer „celý text odzadu“ a „po slovech odzadu“ byl postup šifrování a dešifrování shodný, u šifer „podle plotu“ jde o různé algoritmy. Pokud by Vás zajímal skript pro dešifrování „podle plotu“, můžete se podívat na zdrojový kód webové stránky [2].

V této části seriálu jsme důkladně rozebrali algoritmy šifrování pomocí několika vybraných snadných transpozic. K zápisu programů jsme zvolili skriptovací jazyk JavaScript, jednak pro jeho dostupnost a jednak pro jeho syntaxi, která je velmi podobná syntaxi široce rozšířeného programovacího jazyka C. V příštích částech se seznámíme s dalšími transpozičními šiframi a také s postupy jejich luštění.

## Literatura


[1] *Eisenmenger, R.*: JavaScript, kompletní kapesní průvodce. Přeložila Kamila Slavíčková. 1. vyd. Praha: Grada Publishing, s.r.o. 1999. 304 str. ISBN 80-7169-383-9.

[2] *Musílek, M.*: Šifry – jednoduché transpozice [online]. 2010 [cit. 2010-09-25]. Dostupné z: <http://www.musilek.eu/michal/sifry-poradi.html?menu=cc>.

[3] *Singh, S.*: Kniha kódů a šifer. 2. vyd. Praha: Argo a Dokořán, 2009. 384 s. ISBN 978-80-7363-268-7 (Dokořán), ISBN 987-80-257-0144-7 (Argo).

[4] *Vondruška, P.*: Kryptologie, šifrování a tajná písma. 1. vyd. Praha: Albatros, 2006. 344 s. ISBN 80-00-01888-8.


# Supersonicman — jeden projekt v rámci medzipredmetových vzťahov informatiky a fyziky


*JÁN BEŇAČKA*

Fakulta prírodných vied, Univerzita Konštantína Filozofa, Nitra, SLOVENSKO



Článok prezentuje model pádu človeka vo vzduchu, ktorého účelom je zistiť, či je možné pri páde dosiahnuť nadzvukovú rýchlosť. Model je založený na numerickom riešení pohybovej rovnice Eulerovou metódou v Exceli. Princíp riešenia a jeho implementácia sú ľahko pochopiteľné už na úrovni gymnázia. Programovanie nie je použité. Študenti pri vytváraní modelu získajú nové zručnosti v práci s tabuľkovým kalkulátorom a nové vedomosti z danej problematiky.