

Obsah

1	Sedmero hříchů programátora	1
2	Konstanty	2
3	Soubory	3
3.1	Typy souborů	3
3.2	Funkce pro práci se soubory	4
3.3	Příklady práce se soubory	4

Sedmero hříchů programátora

1. *GOTO* – za jeho použití by měl programátor automaticky "goto hell".
2. *Globální proměnné* – znepřehledňují program, komunikaci mezi funkcemi/procedurami lze vést přes parametry a návratovou hodnotu.
3. *Nečitelné identifikátory* – použití identifikátoru jako „promenna" nebo „xxx" nebo „abc".
4. *Neinicializované proměnné* – proměnné nemají (v mnoha případech) předem danou hodnotu.
5. *Neuniverzální funkce/procedury* – fce, které dělejí nejen to, co mají.
6. *Délka funkce/procedury přes jednu obrazovku (cca 30 řádků)* – ztráta kontextu.
7. *Nepoužívání indentace* – velké snížení přehlednosti programu.

Jak to může skončit, pokud budete páchat hříchy???

[illegible]

```
#include <stdio.h>

#define l11l 0xFFFF
#define l1l for
#define l111l if
#define l1l1 unsigned
#define l11l struct
#define l111l short
#define l111l long
#define l111l putchar
#define l11l(1) l=malloc(sizeof(l11l l111l));l->l11l1=1-1;l->l111l=1-1;
#define l11l *l111l++=l11l%10000;l11l/=10000;
#define l11l l111l(!11->l11l1){l11l(11->l11l1);l1->l11l1->l11l1=1;}\
l11l=(11-11->l11l1)->l1l;l1=1-1;
#define l11l 1000
```

```

11111 11111 {
1111 11111 *
11111 1111 {
11111 [1]
11111;};main
(){}1111 11111
11,* 1111,*
11,* malloc ( ) ; 1111
1111 1111,11 ,1;1111 1111 *1111,*
-1-1; <1<; 1111("vT'8")>1V'9?,"w1"
);scanf("%d",&l);1111(111) 1111(111
11[111->111[1-1] =1;1111;111(111
+*111)(11->1111;
111; 11111 =(
);111(111->
+*111**1111++
1111 1111=(
);111(111;
{ 1111 } *
111(1=(11->1
(1->11)[ 1
11->1111,1=
+011)printf(
19,"%n%04d")

```

3 Soubory

3.1 Typy souborů

Typy souborů v Pascalu

Pascal rozeznává 3 druhy souborů

- Textový soubor
 - Práce podobná jako s klávesnicí a obrazovkou
 - Obsahuje text organizovaný do řádků
 - Datový typ `text`
- Typový soubor
 - Obsahuje položky s daným typem
- Netypový soubor
 - Volný formát souboru

Typový soubor

- Pevná délka záznamu v souboru
- Datový typ `file of type`
- K datům se přistupuje po přesně daných „kvantech“
- Podobným způsobem funguje datový formát DBF (dBase, ...)

Netypový soubor

- Volný přístup k datům v souboru
- Data nemají (resp. nemusí mít) žádný jednoduchý datový formát
- Datový typ `file`

3.2 Funkce pro práci se soubory

Funkce pro práci se soubory

- `Assign` – přiřazení proměnné souboru
- `Append` – otevření souboru pro vkládání na konec
- `Reset` – otevření existujícího souboru
- `Rewrite` – vytvoření (příp. i smazání) a otevření nového souboru

- Close – uzavření souboru
- Read/ReadLn/BlockRead – čtení ze souboru
- Write/WriteLn/BlockWrite – zápis do souboru
- Eof/Eoln – zjištění konce souboru/řádku
- SetTextBuf – nastavení vstupně/výstupního bufferu
- Flush – vyprázdnění vstupně/výstupního bufferu
- FileSize – zjištění velikosti souboru
- FilePos – zjištění pozice v souboru
- Seek – nastavení pozice v souboru
- IOResult – test úspěšnosti práce se souborem

3.3 Příklady práce se soubory

Vytvoření nového textového souboru

```
var f:text;
begin
  Assign(f, 'soubor.txt');  Přiřazení jména
  Rewrite(f);               Přepsání souboru
  ...                       Práce se souborem
  Close(f);                 Uzavření souboru
end.
```

Otevření textového souboru pro čtení

```
var f:text;
begin
  Assign(f, 'soubor.txt');  Přiřazení jména
  Reset(f);                 Otevření souboru
  ...                       Práce se souborem
  Close(f);                 Uzavření souboru
end.
```

Otevření textového souboru pro přidávání

```
var f:text;
begin
  Assign(f, 'soubor.txt');  Přiřazení jména
  Append(f);                Otevření souboru
  ...                       Práce se souborem
  Close(f);                 Uzavření souboru
end.
```

Otevření textového souboru pro přidávání II.

```
var f:text;
begin
  Assign(f, 'soubor.txt'); Přiřazení jména
  {$I-} Vypnutí běhových chyb
  Append(f); Otevření souboru
  {$I+} Zapnutí běhových chyb
  if IOResult<>0 then Nepovedlo se otevřít
    Rewrite(f); Vytvoření nového souboru
    ... Práce se souborem
    Close(f); Uzavření souboru
end.
```

Zápis a čtení textového souboru

```
... Otevření pro zápis
WriteLn(f, 'Ahoj svete!'); Zápis do souboru
Reset(f); Znovuotevření pro čtení
ReadLn(f, s); Čtení ze souboru
```