Instructiunea de selectie (switch)

Forma generală a acestei instrucțiuni este:

```
switch( <exp_int> )
{
        case <exp_ct_1>: <secventa_1>
            //.............
        case <exp_ct_k>: <secventa_k>
        default: <secventa_x>
}
```

Componentele de bază ale instrucțiunii de selecție au următoarele particularități:

- expresia selectoare **<*exp_int*>** trebuie să aibă valoarea de tip întreg;
- etichetele ***case* <*exp_ct_i*>**, cu *i*=1,..,*k* trebuie să aibă în partea **exp_ct_i** constante sau expresii cu valoare constantă, de tip întreg; ele trebuie să fie distincte;
- eticheta **default** şi prelucrarea asociată ei sunt opționale; ea corespunde mulțimii etichetelor posibile diferite de cele menționate explicit în case,
- **<secventa_i>** reprezintă instrucțiuni simple sau compuse, dar pot şi lipsi. In cazul acestor secvente de program poate aparea si instructiunea ***break*,** efectul fiind acela de a parasi instructiunea switch, in acel punct si continuarea programului cu urmatoarea instructiune de dupa switch**.** Construcții ca urmatoarea fiind perfect valide:

OBSERVATIE

Acoladele din cadrul instrucțiunii ***switch*** sunt absolut necesare, deoarece delimitează instrucțiunea compusă asociată.

Instructiunea de selectie (switch)

```
switch ( nota )
{
    case 10 :
    case 9  :printf ("\nFelicitari !");
             break;
    case 8  :
    case 7  :printf("bine")
             break;
    case 6  :printf("...");
             break;
    case 5  :printf(" ? ");
             break;
    default :printf("\nNota inacceptabila.");
}
```

Derularea execuției unei instrucțiuni de selecție:

- se evaluează expresia selectoare, a cărei valoare este comparată succesiv cu valorile **<*exp_ct_i*>** din etichetele case;
- dacă se detectează o etichetă având pentru **<*exp_ct_i*>** o valoare egală cu cea a expresiei **<*exp_int*>**, se trece la execuția prelucrării asociate, continuându-se apoi cu toate cele care urmează, până la întâlnirea unei instrucțiuni ***break***, ***return*** sau până la întâlnirea acoladei de sfârşit a instrucțiunii complexe asociate.
- dacă nu se detectează o astfel de etichetă, dar este prezentă eticheta ***default***, se execută prelucarea asociată acesteia, altfel selecția se încheie fără a fi executată vreo prelucrare.

Instructiunea de selectie (switch)

OBSERVATIE

Este important de subliniat că execuția instrucțiunilor care încep de la o anumită etichetă case continuă până la întâlnirea unei instrucțiuni *break* sau până la întâlnirea ultimei acolade din instrucțiunea complexă asociată lui *switch*.

Astfel, secventa:

```
if( expresie==n1 )
        <prelucrare_1>
else if( expresie==n2 )
        <prelucrare_2>
         ..........
else if( expresie==nk )
        <prelucrare_k>
else    <prelucrare_x>
```

Nu este echivalenta cu ***switch***, decat daca adaugam ***break*** dupa fiecare <*prelucrare_i*>

```
switch( expresie )
{       case n1: <prelucrare_1>//break;
        case n2: <prelucrare_2>//break;
        //..........
        case nk: <prelucrare_k>//break;
        default: <prelucrare_x>//break;
}
```

Instructiunea de selectie (switch)

EXEMPLU

Programul va citi un şir de caractere şi-l va reafişa în următoarele condiţii:

- **caracterul '\r' va fi afişat precedat de şirul "<CR>";**
- **caracterul TAB va fi înlocuit cu şirul "->";**
- **cifra zero, pentru a nu fi confundată cu litera o, va fi înlocuită cu şirul <zero>;**
- **celelalte caractere din şir vor fi afişate aşa cum au fost citite.**

```
/*
    Program de înlocuire a caracterelor:
'\r' cu sirul <CR> si apoi salt la rând nou
'\t' cu sirul ->, fara a mai afisa TAB
cifra 0 cu sirul <zero> .
    Programul se incheie atunci cind se tasteaza ESCAPE.
*/
#include <stdio.h>
#include <conio.h>

#define  ESC    0x1b
#define  TAB    '\t'
```

Instructiunea de selectie (switch)

```
int main( void )
{
    char c;
    //clrscr();
    printf ("\nIntroduceti un sir terminat cu ESC:\n");
    while( (c=getch()) != ESC )
        switch( c )
        {
         case '\r':
            printf ("<CR>"); putch ('\n');
            putch( c );//echivalent cu: printf("%c", c );
            break;
         case TAB :
            printf ("->");
            break;
         case '0' :
            printf ("<zero>");
            break;
         default :
            putch( c );//echivalent cu: printf("%c", c );
            //break;
        }
    printf("Escape");
    return 0;
}
```

```
Introduceti un sir terminat cu ESC:
<CR>
<CR>
Acesta->este un sir de lungime <zero><CR>
<CR>
Escape
Process returned 0 (0x0)   execution time : 21.052 s
Press any key to continue.
```

Instructiunea de selectie (switch)

EXPLICATII

Instrucțiunea ***switch*** este singura instrucțiune a ciclului ***while***. Din această cauză nu a fost nevoie de acolade suplimentare.

Pentru etichete s-au utilizat, fie constante explicite, '0', '\r', fie simbolice TAB, neexistând restricții în acest sens. Important este să fie o constantă întreagă sau reductibilă la un întreg.

Programarea acestui switch a fost făcută în *manieră preventivă*, astfel încât adăugarea de noi etichete care să nu necesite completarea sau modificarea liniilor deja scrise.

Este adevărat că aceleaşi funcții sunt realizate şi de instrucțiunile:**.**

```
while( (c=getch())!=ESC )
    switch (c)
    {case TAB :
        printf ("->");
        break;
     case '0' :
        printf ("<zero>");
        break;
     case '\r':
        printf ("<CR>");
        putch ('\n');
     default :
        putch (c);
    }
```

Instructiunea de selectie (switch)

EXPLICATII

Se observă însă că prin modificarea locului etichetei ***case*** '\r' programul devine mai scurt cu trei instrucțiuni:

- s-a eliminat putch(c);
- şi o instrucțiune break,
- iar după default nu s-a mai pus break deoarece execuția instrucțiunii switch se opreşte oricum la ultima acoladă.

Aceste modificări nu afectează funcțiile programului. Imaginați-vă însă că trebuie să introduceți o nouă etichetă **case** astfel încât, de exemplu, litera 'o' să fie afişată **<litera>o**. Pare foarte atrăgător să introducem un **printf** imediat înainte de **default** pentru a beneficia de afişarea caracterului 'o' memorat în variabila **c**, adică o modificare efectuată în modul de mai jos:

```
        //...
        case 'o' : printf ("<litera>");
        default  : putch (c);
       }
```

Insă această introducere va avea ca efect alterarea afişării caracterului '\r' care acum va fi afişat prin **<CR><litera>** urmat de un salt la linie nouă.

Plasarea lui ***case*** 'o' după ***default***, fără a se introduce un ***break*** va avea ca efect afişarea şirului "<litera>" după fiecare caracter diferit de TAB sau de cifra zero.

Programarea preventivă, prin care toate secvențele din ***switch*** sunt independente şi se termină cu un ***break***, va permite evitarea unor astfel de necazuri.

2009/2010 - Suceava

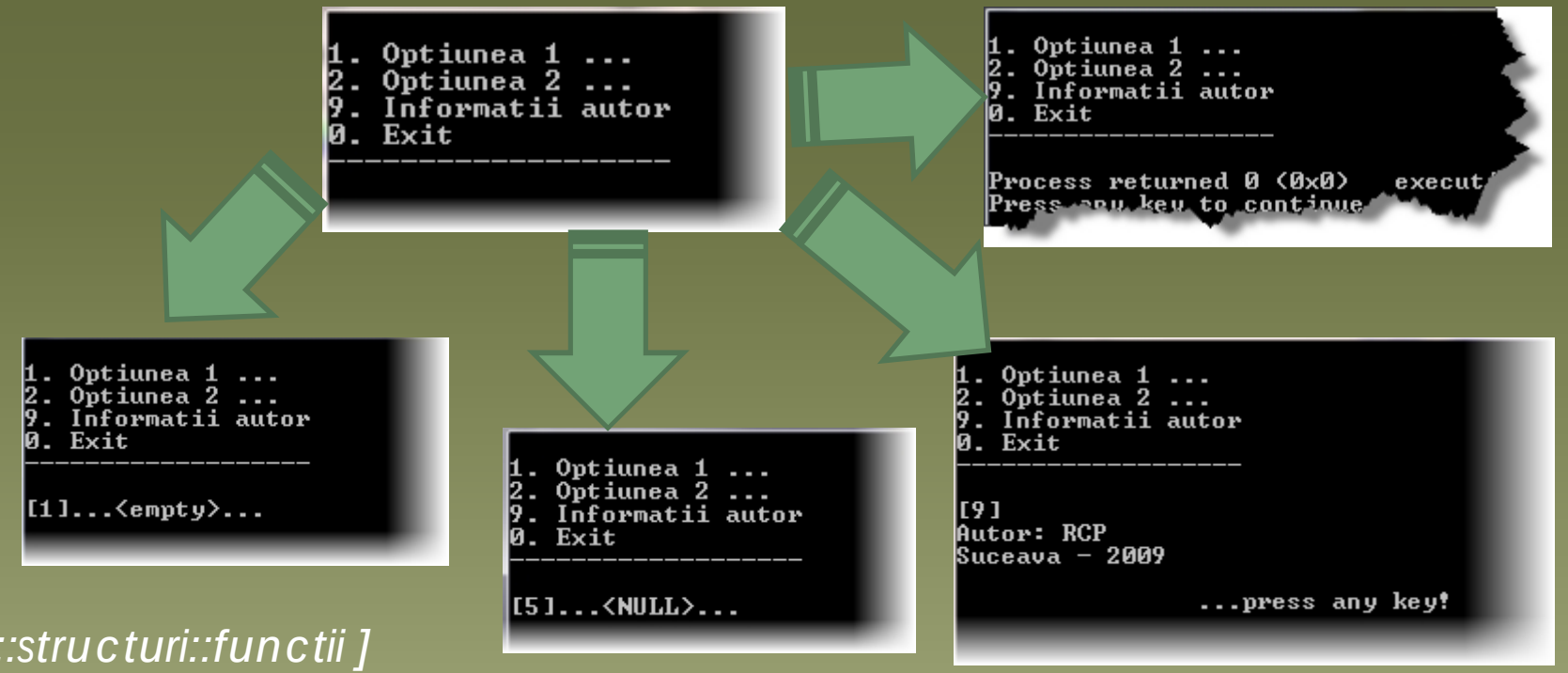
Aplicatii de tip MENU

Consta in crearea de programe capabile sa afiseze o lista de optiuni, etichetate prin diferite simboluri (litere, cifre, etc).

Utilizatorul va putea alege, la un moment dat, o eticheta, prin apasarea unei taste.

In acest moemnt programul va executa o secventa de instructiuni, specifice acelei optiuni, urmand ca la finalul acestora sa revina in modul de afisare a optiunilor.

Una dintre optiuni va fi de genul “X. Parasire program”, optiune ce va avea ca sarcina efectiva, parasirea aplicatiei.



Aplicatii de tip MENU

```
#include <stdio.h>
#include <conio.h>

int main( void )
{   char c;
    do
    {   //echivalent ca efect cu ...
        system("cls");  //clrscr();
        printf( "\n1. Optiunea 1 ..."
                "\n2. Optiunea 2 ..."
                //....
                "\n9. Informatii autor"
                "\n0. Exit"
                "\n--------------------\n" );

        //INSTRUCTIUNEA SWITCH (pe slide-ul urmator) – de inlocuit

}while( !0 ); //bucla infinita
    return -1; //nu este normal sa ies pe aici
}//END main( ... )
```

Aplicatii de tip MENU

```
//...
switch( c = getch() )
{case '0': exit( 0 );
           //break;
 case '9': printf( "\n[%c]\nAutor: RCP"
            "\nSuceava - 2009"
            "\n\n\t\t...press any key!", c );
           getch();//asteapta o tasta
           break;
case '1':  printf( "\n[%c]...<empty>...", c );
           getch();//asteapta o tasta
           break;
case '2':  printf( "\n[%c]...<empty>...", c );
           getch();//asteapta o tasta
           break;
default :  printf( "\n[%c]...<NULL>...", c );
           getch();//asteapta o tasta
           break;
}
 //...
```

2009/2010 - Suceava

Aplicatii de tip MENU. PROBLEME

1. Scrieți un program care va citi notele obținute de o grupă la un examen şi va efectua prelucrările corespunzătoare meniului de mai jos:
   - N - citire note
   - M - Calculul mediei generale a grupei
   - P - calculul procentajului de promovați
   - E - Exit

2. Să se scrie un program care execută operații asupra unui vector conform meniului următor:
   - C - Citire vector
   - P - Calculul mediei aritmetice a numerelor pozitive
   - E - Exit

3. Scrieți un program care afişează meniul de mai jos şi execută prelucrările corespunzătoare:
   - C - Citirea a doi vectori de aceeaşi dimensiune
   - S - calculul produsului scalar
   - A - Afişarea celor doi vectori
   - E - Exit

4. Scrieți un program care citeşte sumele încasate de un mic magazin şi efectuează prelucrările din meniul de mai jos:
   - C - Citire sume
   - T - Calcul TVA (18% din total)
   - M - Cea mai mare sumă încasată de la un client
   - E - Exit

5. Să se scrie programul care va citi un şir de numere întregi, va afişa următorul meniu şi va efectua prelucrarea cerută de utilizator:
   1. Numărul elementelor nule
   2. Minimul elemetelor pozitive
   3. Numărul elementelor pare de pe poz. impare
   4. Exit