

XII. Warunek wielokrotnego wyboru switch ... case

12.1. Gdy mamy więcej niż dwie możliwości

Do tej pory poznaliśmy warunek `if ... else ...`. Po co nam kolejny? Trudno powiedzieć, ale na pewno nie po to, żeby Cię zniechęcać do programowania. Moim zdaniem warunek `switch ... case` został wprowadzony, w celu poprawienia czytelności kodu. Nie umożliwia on bowiem bardziej zaawansowanych warunków, niż oferuje nam to wcześniej omawiany warunek typu `if ... else`. Można nawet powiedzieć, że warunek `switch ... case` ma dużo mniejsze możliwości, od `if ... else`. Po co więc nam on? Otóż są sytuacje, w których warto skorzystać warunku `switch ... case` i dlatego zostanie on omówiony. Przyjrzyjmy się najpierw składni.

```
switch(zmienna)
{
    case wartosc1:
        //tu instrukcje zostaną wykonane jeśli (zmienna==wartosc1)
        break;//koniec warunku; wychodzi z warunku switch

    case wartosc2:
        //tu instrukcje zostaną wykonane jeśli (zmienna==wartosc2)
        break;//koniec warunku; wychodzi z warunku switch

    //tu mogą być kolejne case ...

    default://instrukcja warunkowa switch wykonuje ten kod
        //wtedy i tylko wtedy, gdy nie został spełniony
        //żaden inny wyżej wymieniony warunek
        break;//wychodzi z warunku switch;
}
```

Warto w tym miejscu wymienić kilka przykładów, gdzie warto stosować warunek typu `switch...case` - są to głównie listy wyboru, gdzie na podstawie np. naciśniętego klawisza, wybieramy dalszy przebieg programu. Można również porównywać napis wprowadzony przez użytkownika i gdy któryś z nich będzie taki sam, jak w warunku `switch...case`, to wykona się określony kod.

12.2. O czym należy pamiętać korzystając ze switch ... case

Zostały jeszcze dwie ważne sprawy, które powinieneś wiedzieć jako przyszły programista. Sprawa pierwsza to: w warunku `switch...case` nie można deklarować zmiennych. Jeśli potrzebujesz koniecznie zmienną tymczasową, to musisz ją zadeklarować przed warunkiem `switch...case`. Druga sprawa: jeśli zapomnisz słowa kluczowego `break;`, każdy kolejny warunek się wykona w switch'u (aż do napotkania słowa `break;`).

12.3. Praktyczny przykład wykorzystania switch ... case

Poniżej zamieszczam przykład prezentujący działanie `switch...case`.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    float a;
    float b;
    cout<<"Podaj pierwsza liczbe: ";
    cin>>a;
```

```

cout<<"Podaj druga liczbe: ";
cin>>b;
cout<<"liczba a: "<<a<<endl;
cout<<"liczba b: "<<b<<endl;
cout<<"Menu wyboru:"<<endl;
cout<<"[1] Oblicz: a+b"<<endl;
cout<<"[2] Oblicz: a-b"<<endl;
cout<<"[3] Oblicz: a*b"<<endl;
cout<<"[4] Oblicz: a/b"<<endl;
cout<<"Wpisz numer: ";
int wybor;
cin>>wybor;
cout<<endl<<"Wpisales numer: "<<wybor<<endl;
switch(wybor)
{
    case 1:
        cout<<"Opcja "<<wybor<<" wynik dodawania a+b="<<a+b<<endl;
        break;
    case 2:
        cout<<"Opcja "<<wybor<<" wynik odejmowania a-b="<<a-b<<endl;
        break;
    case 3:
        cout<<"Opcja "<<wybor<<" wynik mnozenia a*b="<<a*b<<endl;
        break;
    case 4:
        cout<<"Opcja "<<wybor<<" wynik dzielenia a/b="<<a/b<<endl;
        break;
    default:
        cout<<"Opcja "<<wybor<<" taka opcja nie istnieje!"<<endl;
        break;
}
getch();
return(0);
}

```

12.4 enum – typ wyliczeniowy

Jest innym spojrzeniem na stałe(**const**). Najlepiej używać ich jako stałych symbolicznych, chociaż możemy z nich tworzyć nowe typy(jednak nie w pełni). Przykład wykorzystania enum:

```

#include <conio.h>
#include <iostream>
int main()
{
    using namespace std;
    enum DDT {Admin, Moderator, Uzytkownik};
    int Piotr = 0;
    int Patzick = 1;
    int Dante = 2;

    //własność enum
    if (Piotr == Admin)
        cout << "Adminem DDT jest Piotr.\n";
    if (Patzick == Moderator)
        cout << "Adminowi pomaga kilku moderatorow\n"
        << "jednym z nich jest Patzick.\n";
}

```

```

if (Dante == Uzytkownik)
cout << "Jednak forum nie istniałoby"
<< " bez swoich uzytkownikow\n\n";

cout << "Oto stale enum-a:\n"
<< DDT(0) << ", "
<< DDT(1) << ", "
<< DDT(2) << ".\n";
cout << "To tez stale enum-a:\n"
<< Admin << ", "
<< Moderator << ", "
<< Uzytkownik << ".";

getch();
return 0;
}

```

Typ ten jest promowany do **int** dla każdej instrukcji (if, for, while, swich). Oznacza to ni mniej ni więcej, iż po prostu stałe **enumerators** są zamieniane na liczby. Co potwierdza prezentowany kod. W naszym kodzie **DDT** staje się typem wyliczeniowym, a nazwy **Admin**, **Moderator**, **Uzytkownik** jego enumeratorami (czyli stałymi symbolicznymi).

Domyślnie tak jako to ma miejsce w tablicach, stałe te są numerowane od 0. Dlatego **if (Piotr == Admin)** jest prawdziwe.

Numerowanie stałych możemy sami zmienić ręcznie wprowadzając ich wartości np. **enum {Admin = 10, Moderator, Uzytkownik =50};**. W tym wypadku **Moderator** będzie miał wartość 11. Przyjrzymy się teraz jak będzie wyglądać pierwszy przykład z użyciem **enum**.

12.5 swich + enum

```

#include <iostream>
#include <conio.h>
int main()
{
using namespace std;
enum {dodawanie = 1, odejmowanie = 2,
mnozenie = 3, dzielenie = 4};
float a;
float b;
cout<<"Podaj pierwsza liczbe: ";
cin >> a;
cout << "Podaj druga liczbe: ";
cin >> b;
cout << "liczba a: " << a <<endl
<< "liczba b: " << b << endl
<< "Menu wyboru:" << endl
<< "[1] Oblicz: a+b" << endl
<< "[2] Oblicz: a-b" << endl
<< "[3] Oblicz: a*b" << endl
<< "[4] Oblicz: a/b" << endl;
cout<<"Wpisz numer: ";
int wybor;
cin>>wybor;
cout<<endl<<"Wpisales numer: "<<wybor<<endl;
switch(wybor)

```

```

{
  case dodawanie:
  cout<<"Opcja "<<wybor<<" wynik dodawania a+b="<<a+b<<endl;
  break;
  case odejmowanie:
  cout<<"Opcja "<<wybor<<" wynik odejmowania a-b="<<a-b<<endl;
  break;
  case mnozenie:
  cout<<"Opcja "<<wybor<<" wynik mnozenia a*b="<<a*b<<endl;
  break;
  case dzielenie:
  cout<<"Opcja "<<wybor<<" wynik dzielenia a/b="<<a/b<<endl;
  break;
  default:
  cout<<"Opcja "<<wybor<<" taka opcja nie istnieje!"<<endl;
  break;
}
getch();
return(0);
}

```

Program działa tak samo, instrukcja **switch** zyskała trochę na czytelności.

12.6 Ćwiczenia

Zaprojektuj aplikację*, zadanie będzie wymagało sporej wiedzy, wykorzystane jest większość poznanych do tej pory informacji. Dodam, iż użyteczne będą biblioteki `#include "ddtconsole.h"` i `#include <ctime>`. Oto jak wygląda aplikacja:

```

C:\> Dokumentacja na stronie: http://ddt.pl
                          Administrowanie forum DDT:
                          1. Czytaj logi          3. Lista oczekujacych prac
                          2. Usun temat         4. Pauza
                                          5. Koniec

Co wybierasz szefie? 1
16 Styczen 2008 00:01 - atak SYN flooding IP 99.99.99.1
20 maj 2008 14:13 - uzytkownik Macius700 blad logowania
31 lipiec 2008 20:09 - aktualizacja Kursu Allegro
5 sierpień 2008 17:29 - zarejestrowal sie uzytkownik Mapet

Co wybierasz szefie? 2
Tematy przeznaczone do usunienia
1. GroupId = 191          2. GroupId = 163          3. GroupId = 671

Co wybierasz szefie? 3
1. Dokumentacja.        2. Artykuly.
3. Dzial Pliki.         4. Sygnatury zamknienia by Dante.

Co wybierasz szefie? 4
Szefie zrob sobie przerwe, ile potrzebujesz minut?
0.3
Twoja przerwa wyniesie(w sekundach): 18
Niestety przerwa juz minela czas do roboty.

Co wybierasz szefie? _

```

Posiada menu(powstało dzięki `console.ddt`) z wyborem pięciu opcji. Po wybraniu jednej z 5 opcji pokażą się odpowiednie dane. Do przechowywania komunikatów(95% wyświetlanych danych) użyto tablicy trójwymiarowej. Opcje 1 - 3 zawierają tekst jak na zdjęciu wyżej. Opcja 4 robi pauzę, ale użytkownik podaje czas paury w jednostkach minutowych czyli jeśli chcesz 6 s paury(to trzeba wpisać 0.1). Jednak komunikat o tym ile użytkownik wybrał pokazuje się w sekundach.

Na kolejnym zdjęciu pokazane jest opcja 5 zamykająca aplikację:

```
GA Dokumentacja na stronie: http://ddt.pl

                          Administrowanie forum DDT:
                          1. Czytaj logi                3. Lista oczekujacych prac
                          2. Usun temat                 4. Pauza
                                                              5. Koniec

Co wybierasz szefie? ass
Nie podano cyfry, spróbuj ponownie
Nie podano cyfry, spróbuj ponownie
Nie podano cyfry, spróbuj ponownie
5
Koniec na dzisiaj do jutra?
Aplikacja zakonczy dzialanie za <sek.>5
Aplikacja zakonczy dzialanie za <sek.>4
Aplikacja zakonczy dzialanie za <sek.>3
Aplikacja zakonczy dzialanie za <sek.>2
Aplikacja zakonczy dzialanie za <sek.>1
```

Koniec polega na odliczeniu ok. 5 sekund i zamknięciu aplikacji. Aplikacja sprawdza czy podczas próby wyboru została wpisana cyfra(liczba), jeśli nie to pojawia się komunikat i dopóki nie zostanie naciśnięta liczba nie da się przejść dalej. **WAŻNE** wszystkie komunikaty, które widzisz są zapisane jako łańcuch w tabeli o której wspomniałem(użycie zapisu cout << "Co wybierasz szefie?" jest niedozwolone). Oczywiście to jak Ty wykonasz aplikację, zależy od Ciebie, ja tylko podałem pewne informacje o tym jak ja ją wykonałem. Jeśli zrobisz to inaczej, a otrzymany wynik będzie taki sam(lub podobny) to bardzo dobrze.

Aplikacja nie jest łatwa, jednak stworzenie menu, oraz pierwsze 4 opcje nie powinny sprawić ci większych trudności jeżeli rzetelnie i sumiennie podeszłaś do nauki kursu.

Problem może ci sprawić odliczanie(czasu zamknięcia) przy zakończeniu programu i sprawdzenie czy naciśnięta została cyfra, ale od tego jest forum by naprowadzić Cię na odpowiednie rozwiązanie.