

X. Tworzenie warunków prostych if ... else

10.1. Intuicja i podejmowanie decyzji

W życiu stale podejmujemy decyzje. Jedne są lepsze, inne gorsze jednak zawsze wybieramy jakąś drogę. Komputer również podejmuje decyzje, lecz trochę inne niż my. Komputer postępuje zgodnie z tym, co mu napiszemy. Nie ma własnej woli, nie ma intuicji - to tylko maszyna. Wszystko więc musi być precyzyjnie sformułowane.

10.2. Warunek if ... else ...

Język C++ pozwala nam sterować przebiegiem programu za pomocą warunków. Żeby umieć poprawnie posługiwać się warunkami, należy zapoznać się zarówno ze składnią warunku, jak również z operatorami warunkowymi (operatorzy porównania).

```
/******  
//Składnia warunku w C++  
if(warunek) jedna_instrukcja; //średnik na końcu!  
/******  
if(warunek)  
{//pierwszy blok instrukcji  
  wiele_instrukcji;  
}  
/******  
if(warunek) jedna_instrukcja; //średnik na końcu!  
else jedna_instrukcja; //średnik na końcu!  
/******  
if(warunek)  
{//pierwszy blok instrukcji  
  wiele_instrukcji;  
}  
else  
{//drugi blok instrukcji  
  wiele_instrukcji;  
}  
/******
```

Jeśli warunek będzie spełniony (prawdziwy), wykonuje się pierwszy blok instrukcji. Jeśli natomiast warunek nie został spełniony i istnieje drugi blok instrukcji (rozpoczyna się on słówkiem **else**), to wykonuje się drugi blok instrukcji.

10.3. Operatory warunkowe

```
//Operatory warunkowe w C++  
== //Równe  
!= //Różne  
>= //Większe równe  
<= //Mniejsze równe  
> //Większe  
< //Mniejsze
```

Operatory warunkowe zademonstruję na przykładzie, po czym szerzej je omówię.

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

```

using namespace std;
int main()
{
    int a,b;
    cout<<"Podaj liczbe a: ";
    cin>>a;
    cout<<"Podaj liczbe b: ";
    cin>>b;

    if(a==b)
    {
        cout<<"liczba a jest rowna liczbie b"<<endl;
    }

    if(a>18)
    {
        cout<<"liczba a="<<a<<" jest wieksza od 18"<<endl;
    }else
    {
        cout<<"liczba a="<<a<<" nie jest wieksza od 18"<<endl;
    }

    if(a>b)
    {
        cout<<"liczba a="<<a<<" jest wieksza od liczby b="<<b<<endl;
    }else
    {
        cout<<"liczba a="<<a<<" nie jest wieksza od liczby b="<<b<<endl;
    }
    getch();
    return(0);
}

```

Operator warunkowy, jak można łatwo zaobserwować na podanym przykładzie, służy do porównywania dwóch wartości. W ogólności warunek można zapisać w postaci: **wartość_A operator warunkowy wartość_B**. Za **wartość** możemy podstawić dowolną liczbę, zmienną liczbową, czy też dowolne wyrażenie matematyczne.

10.4. Zadania do przeanalizowania

Poniżej zamieszczam dwa zadania, które powinienś uzupełnić, w celu lepszego zrozumienia dotychczasowego materiału.

```

//Zadanie: Zad06x04.cpp
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    cout<<"Program sluzы do liczenia miejsca zerowego dla rownania liniowego."<<endl;
    cout<<"Funkcja ma postac: f(x)=a*x+b"<<endl<<endl;
    float a,b;
    cout<<"Podaj a: ";
    cin>>a;
    cout<<"Podaj b: ";
    cin>>b;
    //miejsce zerowe to: 0=a*x+b <=> -b=a*x <=> (-b)/a=x
}

```

```

if(...)//tu uzupełnić kod
{
    cout<<"Funkcja f(x) przyjmuje wartosc=0 dla x="<<(-b)/a<<endl;
}else cout<<"Funkcja f(x) nie posiada rozwiazania."<<endl;
getch();
return(0);
}
//Zadanie: Zad06x05.cpp
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    cout<<"Program sluzzy do sprawdzenia, czy rownanie kwadratowe ma miejsca zerowe."<<endl;
    cout<<"Funkcja ma postac: f(x)=a*x^2+b*x+c"<<endl<<endl;
    float a,b,c,delta;
    cout<<"Podaj a: ";
    cin>>a;
    cout<<"Podaj b: ";
    cin>>b;
    cout<<"Podaj c: ";
    cin>>c;
    delta=b*b-4*a*c;
    if(...)//Tu pierwszy warunek
    {
        cout<<"Funkcja posiada dwa miejsca zerowe."<<endl;
    }else
    if(...)//Tu drugi warunek
    {
        cout<<"Funkcja posiada jedno miejsce zerowe."<<endl;
    }else
    {
        cout<<"Funkcja nie posiada miejsc zerowych."<<endl;
    }
    getch();
    return(0);
}

```

10.5 Sprawdź co w zdaniu piszczy.

W C++ istnieje pewna ciekawa biblioteka, która świetnie nadaje się do analizy znaków(char). Zobaczmy jak ona działa i co nam oferuje:

```

#include <conio.h>
#include <iostream>
#include <cctype> //nowa biblioteka
int main()
{
    using namespace std;
    char znak;
    int litery = 0, liczby = 0,
    biale_znaki = 0, z_przestankowe = 0,
    inne_znaki = 0;

    cout << "Wpisz tekst do analizy, by zakonczyc"
    " wpisz znak |\n";

```

```

cin.get(znak);
while(znak != '|')
{ //sprawdza czy litery
  if(isalpha(znak))
  litery++;
  //sprawdza czy liczby
  else if(isdigit(znak))
  liczby++;
  //sprawdza czy białe znaki
  else if(isspace(znak))
  biale_znaki++;
  //sprawdza czy znaki przestankowe
  else if(ispunct(znak))
  z_przestankowe++;
  else
  inne_znaki++;
  cin.get(znak);
}
cout << "\nW Twoim tekście występują:\n"
<< "Litery - " << litery
<< endl
<< "Cyfr - " << liczby
<< endl
<< "Białe znaki - " << biale_znaki
<< endl
<< "Znaki przestankowe - " << z_przestankowe
<< endl
<< "Inne znaki - " << inne_znaki;

getch();
return 0;
}

```

W przykładzie przedstawiono możliwości **pętli i if-ów**, zastosowano ciekawą **bibliotekę ctype**. Info o biblioteki.