

XIII. Polecenia `continue`; `break`;

13.1. Sukcesywna nauka to podstawa sukcesu

Skoro poznaliśmy już wszystkie możliwe pętle, warto w tym miejscu wtrącić słowo o instrukcjach **`continue`**; i **`break`**; . Jeśli nadal masz problemy z posługiwaniem się pętlami, wróć do wcześniejszego materiału ponieważ ten rozdział zakłada już, że znasz dobrze zasady działania każdej pętli.

13.2. Słowo kluczowe `continue`;

Kompilator C++, jak w większości języków programowania umożliwia nam modyfikowanie w dość specyficzny sposób działanie pętli. Słowo kluczowe **`continue`**; użyte wewnątrz pętli powoduje przerwanie wykonywania bloku instrukcji i przechodzi do sprawdzenia warunku kończącego pętlę. Jeśli jest to pętla **`for(...)`**, wykonuje się jeszcze inkrementacja (lub dekrementacja) zmiennej.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int i=0;
    do
    {
        cout<<"i="<<i<<endl;
        if(i==0)
        {
            i+=1;
            continue;
        }
        cout<<"koniec"<<endl;
    }while(i==0);
    for(i=0;i<5;i++)
    {
        cout<<"[for] i="<<i<<endl;
        if(i>2) continue;
        cout<<"[for]koniec"<<endl;
    }
    getch();
    return(0);
}
```

13.3. Słowo kluczowe **break**;

Drugim słowem kluczowym, które modyfikuje działanie każdej pętli to **break**. Użycie tego słowa wewnątrz pętli zmusi ją do natychmiastowego jej przerwania. Słowo kluczowe **break** zatrzymuje pracę tylko jednej pętli. Oznacza to, że jeżeli zagnieździmy dwie pętle, to zostanie zatrzymana pętla ta, w której zostało słowo kluczowe **break**; użyte. Poniższy przykład ilustruje działanie tej instrukcji w praktyce.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int j;
    for(j=0;j<10;j++)
    {
        cout<<"Aktualne j to: "<<j<<endl;
        for(int i=0;i<5;i++)
        {
            cout<<"[for] i="<<i<<endl;
            if(i>2) break;
            cout<<"[for]koniec"<<endl;
        }
        if(j>6) break;
    }
    getch();
    return(0);
}
```

13.4. Wskazana powściągliwość

Zanim zaczniesz nagminnie używać tych poleceń, zastanów się lepiej jak można napisać kod unikając używania słów kluczowych **continue**; i **break**. Praktycznie zawsze istnieje możliwość napisania pętli bez użycia wymienionych słów kluczowych. Używaj ich tylko w skrajnie uzasadnionych przypadkach czyli wtedy gdy naprawdę uważasz, że czytelniejszym rozwiązaniem będzie użycie jednego z wymienionych słów kluczowych, niż modyfikacja pewnej partii kodu.