
	Politechnika Bydgoska im. Jan i Jędrzeja Śniadeckich WYDZIAŁ TELEKOMUNIKACJI, INFORMATYKI I ELEKTROTECHNIKI	
Podstawy programowania - laboratorium	mgr inż. Marta Gackowska	
Laboratorium 2	Instrukcje sterujące – część 1.	

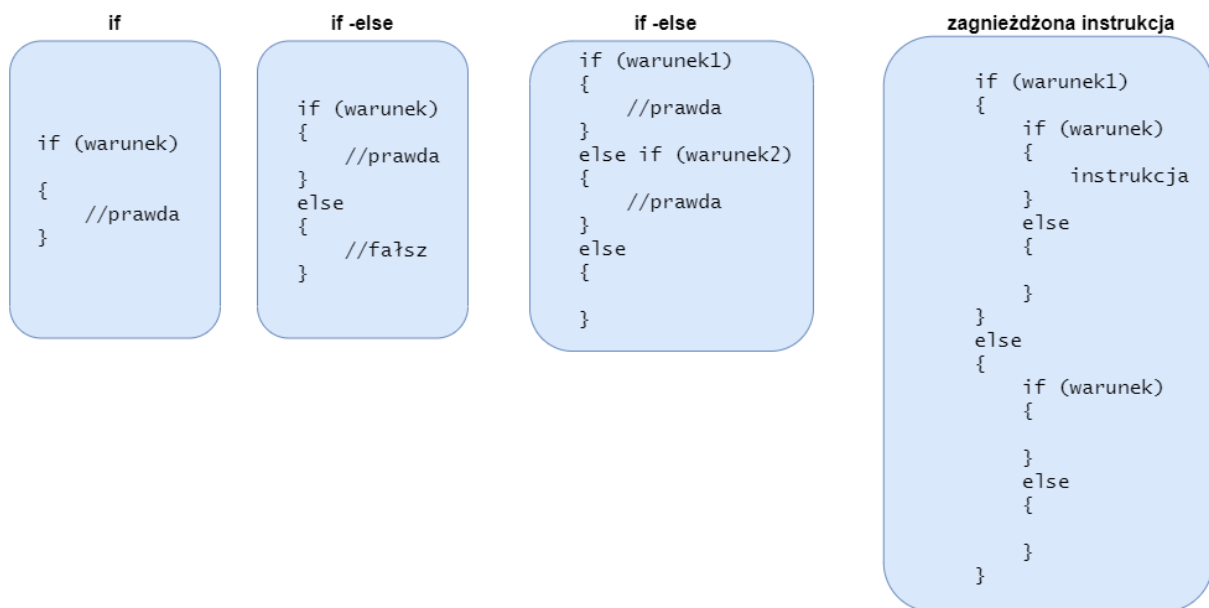
Operatory i operatory relacyjne (puste miejsce na własne notatki)		
<		+
<=		-
==		*
>=		/
>		%
!=		
Operatory przypisania		
+=	x += 20	x = x + 20
-=	rabat -= 2	rabat = rabat - 2
*=	czas *= 3	czas = czas * 3
/=	masa /= 5	masa = masa / 5
%=	redukcja %= 2.45	redukcja = redukcja % 2.45
Operatory logiczne		
&&	i (and)	6 > 2 && 3 == 3
	lub (or)	5 > 2 4 > 7
!	nie (not)	!(4 > 7)
Operatory inkrementacji		
++a	Tzw. tryb przedrostkowy	a zostaje zwiększona przed użyciem jej wartości
a++	Tzw. tryb przyrostkowy	a zostaje zwiększona po użyciu jej wartości
Operatory dekrementacji		
--a	Tzw. tryb przedrostkowy	analogicznie jak wyżej
a--	Tzw. tryb przyrostkowy	analogicznie jak wyżej

Uwaga różnica! Porównanie, a przypisanie.

Kajaki == 5 sprawdza czy zmienna kajaki ma wartość 5 (porównanie)

Kajaki = 5 nadaje zmiennej kajaki wartość 5 (przypisanie)

Operatory inkrementacji i dekrementacji mają bardzo wysoki priorytet wiązania; pierwszeństwo przed nimi mają tylko nawiasy.



Rys.1 Schematy działania instrukcji if, if-else

Instrukcje warunkowe if

Instrukcja warunkowa **if** (tzw. instrukcja alternatywy) sprawdza czy badana zależność (wyrażenie warunkowe) zachodzi czy nie.

Stosowanie klamr zwiększa czytelność kodu, w przypadku coraz bardziej zaawansowanych (złożonych) instrukcji sterujących.

Listing 1

```
#include <stdio.h>
int main() {
    int liczba;
    printf( "Podaj dowolna liczbe calkowita A = ");
    scanf( "%d" , &liczba );
    if (liczba % 2 == 0)
        printf( "Podana liczba jest parzysta.");
    return 0;
}
```

Instrukcja if – else, gdzie else oznacza w przeciwnym razie

Listing 2

```
#include <stdio.h>
int main() {
    int liczba;
    printf( "Podaj dowolna liczbe calkowita A = ");
    scanf( "%d" , &liczba );
    if (liczba % 2 == 0) {
        printf( "Podana liczba jest parzysta.");
    }
    else {
        printf("Podana liczba jest nieparzysta.");
    }
    return 0;
}
```

Listing 3

```
#include <stdio.h>
int main() {
    int liczba;
    printf( "zgadnij liczbe calkowita wieksza niz 10 mniejsza niz 20: \n");
    scanf( "%d", &i);
    if (liczba == 10) {
        printf("ta liczba to 10");
    }
    else if (liczba == 15)
        printf("ta liczba to 15");
    else if (liczba == 20)
        printf("ta liczba to 20");
    else
        printf("Tym razem sie nie udalo.");
    return 0;
}
```

Operator warunkowy ?:

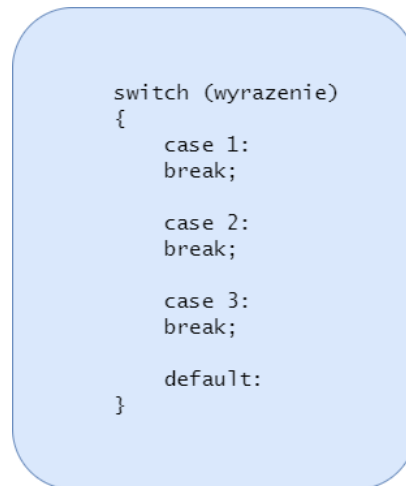
Ogólna postać wyrażenia warunkowego wygląda tak:

wyrażenie1 ? wyrażenie2 : wyrażenie3

Jeśli wyrażenie1 jest prawdziwe (niezerowe), to całe wyrażenie ma wartość taką samą jak wyrażenie2. Jeśli wyrażenie1 jest fałszywe (równe zero), całe wyrażenie warunkowe otrzymuje wartość wyrażenie3.

Switch

Operator warunkowy i konstrukcja if else są łatwymi sposobami realizacji wyboru jednej z dwóch możliwości. Czasem jednak przedmiotem decyzji programu jest kilka wariantów. W takiej sytuacji możemy posłużyć się opisaną wcześniej strukturą if albo if...else jednak w większości przypadków wygodniej jest skorzystać instrukcji switch.



Rys. 2 Schemat działania instrukcji switch

Działanie instrukcji Switch

Na początku obliczane jest wyrażenie w nawiasie po słowie kluczowym switch. Następnie program wykonuje skok do wiersza zawierającego etykietę (etykietami są case np. case 1) odpowiadającą wartości wyrażenia. Jeśli nie ma etykiety wówczas program próbuje wykonać skok do wiersza oznaczonego default. Jeśli taki wiersz nie istnieje program przechodzi do instrukcji następującej po strukturze Switch. Z kolei instrukcja break powoduje opuszczenie struktury Switch i przeskoczenie do pierwszego wiersza który po niej następuje. Gdyby jej nie było po skoku do etykiety program wykonywałby wszystkie instrukcje do samego końca bloku Switch, włącznie z instrukcjami należącymi do innych etykiet.

Listing 4

```
#include <stdio.h>
int main()
{
    int liczba;
    printf("\n Podaj liczbę:");
    scanf("%d", &liczba);
    switch( liczba)
    {
        case 0:
```

```

    printf("\n Liczba 0");
    break;
case 1:
    printf("\n Liczba 1");
    break;
case 2:
    printf("\n Liczba 2 ");
    break;
default :
    printf("\n Inna liczba");
}
return 0;
}

```

Zadania

1. Napisz program, który wczyta 4 liczby rzeczywiste z klawiatury, a następnie sprawdzi czy wśród nich jest więcej liczb dodatnich (≥ 0) czy ujemnych (< 0).
2. Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą n i wypisuje na standardowe wyjście wartość bezwzględną z n . Do rozwiązania zadania nie używaj funkcji bibliotecznych za wyjątkiem operacji wejścia/wyjścia.
3. Wykorzystując wyrażenie warunkowe z operatorem warunkowym $?:$ napisz równoważną instrukcję `if else`.

$$x = (y < 0) ? -y : y;$$

4. Napisz program kalkulator, który pobiera od użytkownika 2 liczby poddawane obliczeniom (zmienny przecinek). Użytkownik wybiera operator (+/*), a cała operacja z wynikiem jest wyświetlana.
5. Napisz program, który w zależności od wyboru użytkownika wczytuje ze standardowego wejścia wymiary: kwadratu, prostokąta lub trójkąta i wypisuje na standardowym wyjściu pole figury o wczytanych wymiarach. Wykorzystaj bibliotekę `math.h`

Zadanie 1	Zadanie 2	Zadanie 3	Zadanie 4	Zadanie 5
3 pkt	3 pkt	1 pkt	3 pkt	4 pkt

Co powinno zawierać sprawozdanie:

Treść zadania, screen kodu i programu wynikowego, uzasadnienie wykorzystanych elementów w programie, (co ma wskazywać na zrozumienie działania napisanego programu), wyjaśnienie jak działa program, wnioski końcowe.