Nazwa implementacji: Nauka języka C – wyrażenia warunkowe if-else

Autor: Piotr Fiorek

Opis implementacji: Poznanie struktury oraz zastosowania wyrażeń warunkowych if-else w języku C.

W programie realizującym jakiś algorytm bardzo często to, co program ma zrobić dalej, zależy od jakiejś zmiennej – zależnie od jej wartości program zachowa się w jeden albo drugi sposób.

Do tworzenia takiego zachowania służy instrukcja warunkowa „if”.
Instrukcja ta sprawdza wyrażenie logiczne i zależnie od tego czy jego wynik jest prawdziwy, czy nie wykonuje instrukcje, albo nie. Wygląda to tak:

```
if(wyrażenie)

{

instrukcja1;

instrukcja2;

...

}
```

Zamiast „wyrażenie” podstawiamy dowolne wyrażenie zwracające wartość prawda lub fałsz jak np. „zmienna1 > 5”, „zmienna1 == 3”, „zmienna1 != 0” lub dowolne inne. Warto zwrócić uwagę na to, że porównanie w C wykonujemy poprzez wstawienie podwójnego znaku równości. Pojedynczy jak widzieliśmy wcześniej, jest operatorem przypisania wartości do zmiennej. Należy o tym pamiętać, ponieważ jeśli wstawimy pojedynczy znak równości, to wyrażenie będzie prawdziwe, jeśli tylko uda się do zmiennej przypisać wartość, a z reguły się to udaje. Nierówność, jak widać w przykładzie, zapisujemy jako „!=”.

Warto też w tym miejscu powiedzieć, że w C zero jest rozpoznawane jako fałsz, a wszystkie pozostałe wartości jako prawda.

Instrukcja „if” może mieć też opcjonalną część „else” oznaczającą „w przeciwnym wypadku”. W ten sposób nasz program może np. sprawdzić czy zmienna jest równa zero i jeśli tak to wykonać jakąś czynność, a jeśli nie to zrobić coś zupełnie innego:

```
if(wyrażenie)

{

instrukcja1;

instrukcja2;
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
...
}
else
{
instrukcja1;
instrukcja2;
...
}
```

Przykładowy program może wyglądać tak:

```
#include <stdio.h>
int main(void)
{
int zmienna;
printf("Podaj wartosc zmiennej: ");
scanf("%d", &zmienna);
if(zmienna > 10)
{
printf("Zmienna ma wartosc %d i jest wieksza niz 10\n", zmienna);
}
else
{
printf("Zmienna ma wartosc %d i jest mniejsza niz 10\n", zmienna);
}
return 0;
}
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

W tym programie po kolei: deklarujemy zmienną typu całkowitego, drukujemy tekst zachęcający do wprowadzenia wartości dla zmiennej, pobieramy z klawiatury wartość do zmiennej (nie będę w tej chwili pisał o szczegółach, ale do pobierania wartości z klawiatury służy funkcja „scanf", która przyjmuje parametry tak samo jak funkcja „printf" tylko nazwa zmiennej musi być poprzedzona znakiem „&"), no i korzystając z instrukcji „if ... else" wyświetlamy komunikat i wartość zmiennej. Można też łączyć ze sobą wiele bloków warunkowych, aby budować bardziej złożone warunki:

```
if(wyrażenie)
{
instrukcja1;
instrukcja2;
...
}
else if(wyrażenie)
{
instrukcja1;
instrukcja2;
...
}
...
else
{
instrukcja1;
instrukcja2;
...
}
```

W ten sposób program może rozpoznawać nawet bardzo złożone przypadki i podejmować działania zależnie od potrzeb.

Zmodyfikujmy zatem nasz dotychczasowy program, aby mógł służyć jako prosty kalkulator. Program będzie pytał użytkownika, jaką operację chce wykonać, a następnie prosił o cyfry, na których ma operować:

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt "Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych" współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
#include <stdio.h>

int main(void)

{

int zmienna, cyfra1, cyfra2, wynik;

printf("Jaką operację chcesz wykonać?\n1) dodawanie\n2) odejmowanie\n3) mnożenie\n4) dzielenie\n");

scanf("%d", &zmienna);

printf("Podaj pierwszą cyfrę do operacji: ");

scanf("%d", &cyfra1);

printf("Podaj drugą cyfrę do operacji: ");

scanf("%d", &cyfra2);

if(zmienna == 1)

{

wynik = cyfra1 + cyfra2;

}

else if(zmienna == 2)

{

wynik = cyfra1 - cyfra2;

}

else if(zmienna == 3)

{

wynik = cyfra1 * cyfra2;

}

else if(zmienna == 4)

{

wynik = cyfra1 / cyfra2;

}

else

{
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
printf("Wybrales nieistniejaca opcje\n");

return 1;

}

printf("Wynik wynosi: %d\n", wynik);

return 0;

}
```

Kod programu powinien być całkowicie zrozumiały: Na początku deklarujemy zmienne i wczytujemy wartości do nich. Później na podstawie zmiennej „zmienna” program wybiera, jaką operację wykonać i przeprowadza ją, a na końcu wyświetlany jest wynik. Warto zwrócić uwagę na ostatnią operację „else” – zostanie ona wykonana, jeśli wszystkie inne warunki będą fałszywe. Jeśli zmienna służąca do wyboru operacji nie będzie z zakresu od 1 do 4 to zostanie wyświetlona informacja o wybraniu złej opcji oraz program zakończy działanie z kodem „1” (każdy kod poza „0” zwyczajowo uznawany jest za kod błędu).

Bloki warunkowe można też zagnieżdżać wewnątrz siebie tworząc bardziej szczegółowe decyzje. Dodajmy do naszego dzielenie sprawdzenie czy użytkownik nie próbuje dzielić przez zero:

```
#include <stdio.h>

int main(void)
{
   int zmienna, cyfra1, cyfra2, wynik;

   printf("Jaką operację chcesz wykonać?\n1) dodawanie\n2) odejmowanie\n3) mnożenie\n4) dzielenie\n");
   scanf("%d", &zmienna);

   printf("Podaj pierwszą cyfrę do operacji: ");
   scanf("%d", &cyfra1);
   printf("Podaj drugą cyfrę do operacji: ");
   scanf("%d", &cyfra2);

   if(zmienna == 1)
   {
       wynik = cyfra1 + cyfra2;
   }
   else if(zmienna == 2)
   {
       wynik = cyfra1 - cyfra2;
   }
   else if(zmienna == 3)
   {
       wynik = cyfra1 * cyfra2;
   }
   else if(zmienna == 4)
   {
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
        if(cyfra2 == 0)
        {
            printf("Nie wolno dzielic przez zero\n");
            return 1;
        }
        wynik = cyfra1 / cyfra2;
    }
    else
    {
        printf("Wybrales nieistniejaca opcje\n");
        return 1;
    }

    printf("Wynik wynosi: %d\n", wynik);

    return 0;
}
```

Wygląda to dokładnie tak samo jak normalne używanie bloków warunkowych. Jak widać, nie użyliśmy tutaj opcjonalnej części „else”, ponieważ jeśli warunek będzie prawdziwy to program i tak zakończy działanie więc nie trzeba pilnować, aby pozostały kod nie został wykonany.

Warunek może być oparty na kilku rzeczach. Wyrażenia można ze sobą łączyć za pomocą operatorów logicznych: lub –„||” oraz i – „&&”. W ten sposób, gdybyśmy np. chcieli wykonać jakąś operację tylko jeśli obie zmienna są różne od zera, warunek zapisalibyśmy jako „(zmienna1 != 0) && (zmienna2 != 0)”. W takich przypadkach warto brać wyrażenia cząstkowe w nawiasy, aby nie było wątpliwości, co z czym łączymy.

Zadania:

- **Wczytaj z klawiatury trzy zmienne i wypisz je po kolei od największej do najmniejszej (bonus: program powinien uwzględniać możliwość, że liczby są równe).**

Napisz program wczytujący trzy zmienne i sprawdzające czy z odcinków o takich długościach da się zbudować trójkąt.

Napisz program, który sprawdza czy wpisana liczba jest liczbą parzystą i wypisująca informację czy jest czy nie.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI ORE FWIOO UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.