Nazwa implementacji: Nauka języka C – funkcje

Autor:

Piotr Fiorek

Opis implementacji: Poznanie jak w Pythonie tworzyć własne funkcje oraz kiedy należy program dzielić na funkcje.

Funkcje są wygodnym sposobem dzielenia kodu na kawałki. Stosuje się je, aby podzielić kod na logiczne fragmenty, co zwiększa jego czytelność i ułatwia wprowadzanie poprawek oraz, aby nie musieć kopiować fragmentów kodu wiele razy wszędzie tam, gdzie powtarza się wykonanie jakiejś czynności.

Funkcje zapisujemy poza innymi funkcjami i przed innymi funkcjami, które je wykorzystują. Czasami ze względu na wzajemne wywołania nie jest to możliwe dlatego na początku pliku można umieścić tak zwane deklaracje funkcji, czyli linijki mówiące programowi, że taka funkcja gdzieś dalej istnieje i żeby na razie nie przejmował się tym, że nie zna jej treści. Deklaracja funkcji różni się od samego zapisu funkcji tylko tym, że nie posiada bloku kodu, który stanowi właściwą treść funkcji, tylko sam nagłówek:

```
zwracany-typ-danych nazwa-funkcji(typ-parametru parametr1, typ-parametru parametr2, ...);
```

a prawdziwy przykład deklaracji wygląda np. tak:

```
int dodawanie(int x, int y);
```

Tak może np. wyglądać deklaracja funkcji realizującej dodawanie. Co znaczą kolejne fragmenty? Funkcja przyjmuje parametry, wykonuje na nich operacje i najczęściej zwraca wynik swoich działań. Pierwsza część określa typ zwracanych danych, czyli jakiego typu musi być zmienna, aby móc do niej przypisać to, co funkcja zwróci. W naszym przykładzie jest to „int”. Następnie jest nazwa funkcji, która jest używana do wywoływania funkcji, a później w nawiasach oddzielone od siebie parametry wraz z typami, które musimy podać do funkcji i na których funkcja będzie operować.

Zapis funkcji nie posiada średnika na końcu, za to posiada blok kodu, który jest wykonywany podczas wywołania funkcji.

Przeróbmy nasz kalkulator, aby dla każdej operacji używał osobnej funkcji realizującej daną operację:

```
#include <stdio.h>

int dodawanie(int x, int y);

int odejmowanie(int x, int y);

int mnozenie(int x, int y);
```


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.

```
int dzielenie(int x, int y);

int main(void)

{

int zmienna, cyfra1, cyfra2, wynik, wyjscie=0;

char kontynuj;

while(wyjscie == 0)

{

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);

switch(zmienna)

{

case 1:

wynik = dodawanie(cyfra1, cyfra2);

break;

case 2:

wynik = odejmowanie(cyfra1, cyfra2);

break;

case 3:

wynik = mnozenie(cyfra1, cyfra2);

break;

case 4:

if(cyfra2 == 0)

{

printf("Nie wolno dzielic przez zero\n");
```


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.

```
return 1;

}

wynik = dzielenie(cyfra1, cyfra2);

break;

default:

printf("Wybrales nieistniejaca opcje\n");

return 1;

}

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

do

{

printf("\nCzy chcesz wykonac kolejne obliczenie? (t/n): ");

getchar();

scanf("%c", &kontynuj);

if(kontynuj == 'n')

{

wyjscie = 1;

}

else if(kontynuj != 't')

{

printf("Podales bledna litere!\n");

}

} while((kontynuj != 't') && (kontynuj != 'n'));

}

return 0;

}

int dodawanie(int x, int y)
```


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.

```
{

int suma;

suma = x + y;

return suma;

}

int odejmowanie(int x, int y)

{

int suma;

suma = x - y;

return suma;

}

int mnozenie(int x, int y)

{

int suma;

suma = x * y;

return suma;

}

int dzielenie(int x, int y)

{

int suma;

suma = x / y;

return suma;

}
```

Na początku zamieściliśmy deklaracje funkcji, aby kompilator wiedział, jak wyglądają wywołania funkcji, których używamy, a na końcu zdefiniowaliśmy same funkcje.

Nagłówki funkcji można też tworzyć w osobnych plikach z rozszerzeniem „.h”. W ten sposób jest prościej jeśli później chcemy jakiejś funkcji używać w programie złożonym z kilku plików. Dlatego właśnie na samym początku naszego programu jest linie „#include <stdio.h>”. W tym pliku znajdują się deklaracje funkcji „printf”, „scanf” i „getchar”, których używamy w naszym kodzie.

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.

Należy też pamiętać, że do funkcji przekazywana jest tylko wartość zmiennej, a nie sama zmienna. Ma to znaczenie o tyle, że gdybyśmy chcieli przekazać do funkcji zmienną, w której zapisalibyśmy wynik działania funkcji i w ten sposób chcieli zwrócić go na zewnątrz, to przy takiej konstrukcji jak zaprezentowana takie rozwiązanie nie zadziała. Jak obejść ten problem powiem w rozdziale o wskaźnikach.

Jak wspomniałem na samym początku funkcja

„main”

też jest funkcją, która jest automatycznie wywoływana w momencie uruchomienia naszego programu. Deklaracja funkcji

„main”

mówi, że nasza funkcja zwraca wartość typu

„int”

, a

„void”

(ang. pustka) w nawiasach oznacza, że ta funkcja nie przyjmuje żadnych parametrów. Nie zawsze tak – funkcja

„main”

może przyjmować argumenty, ale nie będziemy o tym mówić w tym podręczniku.

Zadania:

Napisz funkcję, która przyjmuje dwa parametry i zwraca większy z nich.

Napisz funkcję, która jako parametry przyjmuje tablicę oraz jej rozmiar i zwraca największy element z tablicy.

Napisz funkcję, która przyjmuje trzy parametry: tablicę, jej rozmiar i zmienną, i w zmiennej zwraca sumę licz z tablicy (Uwaga: Zadanie należy rozwiązać przy użyciu wskaźników).

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.