Pętle

Piotr Zierhoffer

Institute of Computer Science
Poznań University of Technology

7 października 2012

Jak powtórzyć operację wiele razy?

```
int i = 0;
printf("%d\n", i);
i++;
printf("%d\n", i);
i++;
printf("%d\n", i);
i++;
printf("%d\n", i);
i++;
printf("%d\n", i);
i++;
```

Rodzaje pętli

- `while {...}`
  - dopóki warunek jest spełniony, wykonuj
- `do {...} while`
  - wykonuj, dopóki warunek jest spełniony
- `for {...}`
  - ogólniejsza wersja powyższego

Pętla while

```
while(warunek)
{
   lista operacji;
}
```

Przykład:

```
int i = 0;

while(i < 10)
{
   printf("%d\n", i);
   i++;
}
```

Pętla do.. while

```
do
{
   lista operacji;
} while (warunek);
```

Przykład:

```
int i = 0;

do
{
   printf("%d\n", i);
   i++;
} while(i < 10);
```

Różnice

```c
int i;
scanf("%d", &i);

while( i < 10 )
{
  printf("%d\n", i);
  i++;
}
```

```
> 11

[nic nie wypisuje]
```

```c
int i;
scanf("%d", &i);

do
{
  printf("%d\n", i);
  i++;
} while( i < 10 );
```

```
> 11

11
```

Pętla for

```
for(inicjalizacja; warunek; krok)
{
  operacje;
}
```

Przykład:

```
int i;

for(i = 0; i < 10; ++i)
{
  printf("%d\n", i);
}
```

Porównanie

```
for(inicjalizacja; warunek;
    krok)
{
  operacje;
}
```

```
inicjalizacja;
while(warunek)
{
  operacje;
  krok;
}
```

Podsumowanie

- while

```
int i = 0;

while(i < 10)
{
  printf("%d\n", i);
  i++;
}
```

- for

```
int i;

for(i = 0; i < 10; ++i)
{
  printf("%d\n", i);
}
```

- do.. while

```
int i = 0;

do
{
  printf("%d\n", i);
  i++;
} while(i < 10);
```

Co jeszcze można wrzucić?

break

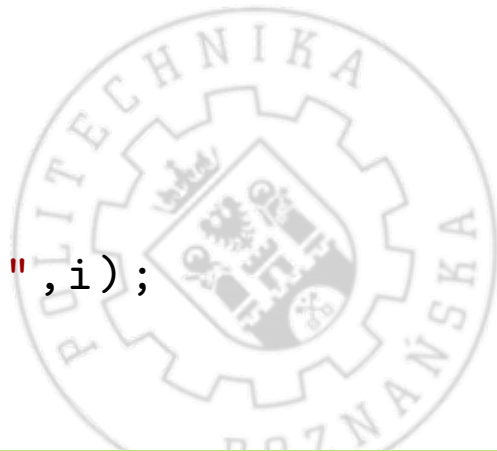
Przerywa wykonywaną pętlę.

```
int i = 0, j;
scanf("%d", &j);

for(i = 0; i < 10; ++i)
{
   pritnf("%d\n",i);
   if( j == i )
   {
      break;
   }
}
```

continue

Przechodzi do następnej iteracji pętli.

```
int i = 0, j;
scanf("%d", &j);

for(i = 0; i < 10; ++i)
{
   if( j == i )
   {
      continue;
   }
   pritnf("%d\n",i);
}
```



Pętle zagnieżdżone

Wszystkie konstrukcje sterujące i warunkowe można zagnieżdżać!

```
int i, j;
int n, m;
scanf("%d␣%d", &n, &m);

for(i = 1; i <= n; ++i)
{
  for(j = 1; j <= m; ++j)
  {
    printf("%d␣*␣%d␣=␣%d\n",
           i, j, i * j);
  }
}
```

```
> 2 3

1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
```

Zadania

Zadanie 1, pętla `while`

Dla wczytanej liczby n, wypisz wszystkie liczby parzyste od 0 do n.

Zadanie 2, pętla `do.. while`

Dla wczytanej liczby `n`, wypisz $\sum_{i=0}^{n} i$.

Zadania

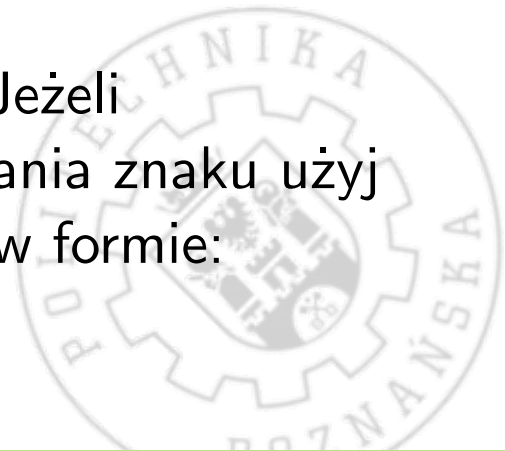
Zadanie 3, pętla `for`

Dla wczytanych liczb n i m (n,m $>$ 0) wypisz gwiazdki w n rzędach i m kolumnach.
Przykład dla n $=$ 3 i m $=$ 4:

```
****
****
****
```

Zadanie 4, pętla `do.. while`

Podaj numer heksadecymalny naciśniętego klawisza. Jeżeli użytkownik naciśnie klawisz ESC, przerwij. Do pobierania znaku użyj funkcji getch() (nagłówek <conio.h>). Wynik podaj w formie:
1 znak to 'a' o kodzie 0x61
2 znak to 'G' o kodzie 0x47



Zadania domowe

Zadanie domowe 1

Napisz algorytm Euklidesa obliczający NWD(a, b), po podaniu przez użytkownika liczb a i b.

Zadanie domowe 2

Napisz prosty kalkulator. Użytkownik wybiera numer opcji z menu jak poniżej, podaje dwie liczby i otrzymuje wynik:

1. Mnożenie
2. Dzielenie //(niecałkowitoliczbowe!)
3. Dodawanie
4. Odejmowanie
5. Modulo
6. Koniec

Po wykonaniu operacji menu pokazuje się ponownie, aż do wyboru opcji "Koniec".