Programowanie strukturalne - wykład 2

dr Piotr Jastrzębski

Język C - dokończenie wprowadzenia

Operator rozmiaru `sizeof()`

Zwraca rozmiar obiektu podany w wybranej jednostce miary, np. bajtach lub słowach maszynowych.

```
printf ("%d",sizeof(int));
```

```
4
```

Operatory bitowe

- negacja bitowa (NOT)(“~”),
- koniunkcja bitowa (AND)(“&”),
- alternatywa bitowa (OR)(“|”) i
- alternatywa rozłączna (XOR) (“^”)

Są zdefiniowane dla liczb całkowitych, działają na bitach i mogą być szybsze niż zwykłe operacje.

https://pl.wikibooks.org/wiki/C/Operatory#Operatory_bitowe

Operatory porównania

- równe (“==”),
- różne (“!=”),
- mniejsze (“<”),
- większe (“>”),
- mniejsze lub równe (“<=”)
- większe lub równe (“>=”)

Operatory logiczne

- negacja (zaprzeczenie): “!”
- koniunkcja (“i”): “&&”
- alternatywa (“lub”): “||”

Operator wyrażenia warunkowego

```
a ? b : c
```

Najpierw oceniana jest wartość logiczna wyrażenia `a`; jeśli jest ono prawdziwe, to zwracana jest wartość `b`, jeśli natomiast wyrażenie `a` jest nieprawdziwe, zwracana jest wartość `c`.

Kolejność operatorów

Operator	Łączność
nawiasy	nie dotyczy
jednoargumentowe przyrostkowe: [] . -> *wywołanie funkcji postinkrementacja postdekrementacja*	lewostronna
jednoargumentowe przedrostkowe: ! ~ + - * & sizeof *preinkrementacja predekrementacja rzutowanie*	prawostronna
* / %	lewostronna
+ -	lewostronna
<< >>	lewostronna
< <= > >=	lewostronna
== !=	lewostronna
&	lewostronna
^	lewostronna
|	lewostronna
&&	lewostronna
||	lewostronna
?:	prawostronna
operatory przypisania	prawostronna
,	lewostronna

Debugowanie

Sesja - live coding.

Instrukcje sterujące

Instrukcje warunkowe

```
if (wyrażenie) {
  /* blok wykonany, jeśli wyrażenie jest prawdziwe */
}
```

```
 if (wyrażenie) {
  /* blok wykonany, jeśli wyrażenie jest prawdziwe */
} else {
  /* blok wykonany, jeśli wyrażenie jest nieprawdziwe */
}
```

```
switch (wyrażenie) {
case wartość1:
 break;
case wartość2:
 break;
/* ... */
default:
 break;
}
```

Pętle

for

```
for (wyrażenie1; wyrażenie2; wyrażenie3) {
   /* instrukcje do wykonania w pętli */
 }
```

do ...while

```
do {
  /* instrukcje do wykonania w pętli */
} while (warunek);
```

while

```
while (warunek) {
   /* instrukcje do wykonania w pętli */
 }
```

Funkcja

Funkcje w C

Ogólna składnia funkcji

```
typ identyfikator (typ1 argument1, typ2 argument2)
{
  /* instrukcje */
}
```

procedura

```
void identyfikator (typ1 argument1, typ2 argument2)
{
  /* instrukcje */
}
```

Rekurencja

Rekurencja, zwana także rekursją (ang. recursion, z łac. recurrere, przybiec z powrotem) – odwoływanie się np. funkcji lub definicji do samej siebie.

$$F_n := \begin{cases} 0 & \text{dla } n = 0, \\ 1 & \text{dla } n = 1, \\ F_{n-1} + F_{n-2} & \text{dla } n > 1. \end{cases}$$

Bibliografia

- https://pl.wikibooks.org/wiki/C/Operatory#Operatory_bitowe, dostęp online 10.03.2020.
- https://pl.wikibooks.org/wiki/C/Funkcje, dostęp online 10.03.2020.
- https://pl.wikipedia.org/wiki/Rekurencja, dostęp online 10.03.2020.
- https://pl.wikibooks.org/wiki/C/Instrukcje_steruj%C4%85ce, dostęp online 10.03.2020.