8. Operátory (C, Java)

Rozdíl unární/binární: unární se vztahuje pouze k jedné proměnné, binární ke dvěma

Příklad: 5 + (-4)

Mínus unární (vztahuje se pouze k 4)

Plus binární (vztahuj se jak k 5 tak k 4)



**Rozdělení:**

Aritmetické (+, -, \*, /, %)

Logické (&&, ||, !)

Bitové (<<, >>, &, |, ^, ~) - jenom zmínit, víc nerozvíjet

Relační (<, >, <=, >=, ==, !=)

Přiřazovací (=, +=, ..)

Ostatní (sizeof)

## **Unární**

Obsah obrázku text, snímek obrazovky, číslo, Písmo

Popis byl vytvořen automaticky

Výsledkem logické negace ! je TRUE nebo FALSE (1,0)

Bitová negace ~ je převrácení bitů výrazu (z 0001 se stane 1110)

### **Operátory inkrementace/dekrementace**

Pokud leží ++ před proměnnou, nejdříve se zvýší její hodnota a pak se proměnná použije ve výrazu (pokud v nějakém je), leží-li ++ za proměnnou, nejdříve se proměnná použije ve výrazu a pak se teprve zvýší její hodnota.



Příklad:

y = 5;

x = ++y; // y se inkrementuje na 6 a poté se dosadí 6 do x

x = y++; // do x se dosadí y (6) a pak se y inkrementuje (na 7)

## **Binární operátory**

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

## **Relační operátory**

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

### **Zkrácený zápis**

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Lze kombinovat: +=, -=, \*-, /=, <<=, >>=, &=, |=, ^=

## **Podmíněný operátor**

Ternární operátor (3 operandy)

Prvním operandem je výraz, který se vyhodnotí jako logický výraz (TRUE nebo FALSE). Pokud se vyhodnotí jako TRUE, výsledkem bude druhý operand (výraz mezi ? a :), jinak třetí operand (výraz za :).



printf("%s\n", (x > y) ? "x je vetsi nez y" : "x je mensi, ci rovno y");



## **Priorita operátorů**

|  |  |  |  |
| --- | --- | --- | --- |
| Priorita | Operátor | Popis | Směr vyhodnocení |
| 0 | () | Závorky/uzávorkování |  |
| 1 | ++ -- | Postfix inkrementace a dekrementace |  |
| () | Volání funkce |
| [] | Přístup k poli |
| . | Přístup ke členům struktury a unionů |
| -> | Přístup ke členům struktur a unionů ukazatelem |
| (typ) {seznam} | Implicitní vytvoření struktur |
| 2 | ++ -- | Prefix inkrementace a dekrementace |  |
| + - | Označení znaménka (int a = -4) |
| ! ~ | Logický NOT a binární negace |
| (type) | Přetypování |
| \* | Dereference pointeru (1) |
| & | Získání adresy |
| Sizeof | Operátor sizeof |
| 3 | \* / % | Násobení, dělení, modulo |  |
| 4 | + - | Sčítání a odečítaní |
| 5 | << >> | Bitový posun |
| 6 | < <= | Porovnávací operátor menší (nebo rovno) |
| > >= | Porovnávací operátor větší (nebo rovno) |
| 7 | == != | Porovnávací operátor je / není rovno |
| 8 | & | Binární AND |
| 9 | ^ | Binární XOR |
| 10 | I | Binární OR |
| 11 | && | Logický AND |
| 12 | || | Logický OR |
| 13 | ? : | Ternární operátor |  |
| 14 | = | Přiřazení |
|  | += (-=, atd..) | Zkrácené zápisy přiřazení |
| 15 | , | Čárka |  |

1. Dereference – přístup k value z paměti