CTRL +  ← / →  – прескача през дума

CTRL + Shift + ← / →  – селектира цяла дума

CTRL + Backspace – изтрива цяла дума зад курсора

CTRL + Del – изтрива цяла дума пред курсора

Shift + ↑ / ↓ – селектира цял ред в зависимост от положението на курсора

Home – праща курсора до началото на реда

End – праща курсора до края на реда

Shift + Home – селектира всичко до началото на реда

Shift + End – селектира всичко до края на реда

CTRL + Home – праща курсора до началото на файла

CTRL + Shift + Home – селектира всичко до началото на файла

CTRL + End – праща курсора до края на файла

CTRL + Shift + End – селектира всичко до края на файла

Ctrl+ . + enter = добавя using Sistem…

class Car – върху Car + ctrl + . -> може да rename-нем навсякъде по кода

Маркиран текст + Ctrl+E+C и Ctrl+K+C = коментар

Ctrl+E+U и Ctrl+K+U = маха коментар

Ctrl+K+D = подравнява кода

Ctrl+. Върху текст подсказва коя библютека ни липсва

I.Логически оператори:

1.Логическо отрицание: сменя стойността на аргумента:

!true = false

!false = true

2. && (и) – Конюнкция – поне единият, ако е равен на false и целият резултат е false (връща истина, когато и двете операнди са истина):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| True | && | False | = | False |
| False | && | True | = | False |
| True | && | True | = | True |
| False | && | False | = | False |

3. || (или) – Дезункция – поне единият от двете операнди е true, разглежда първата операнда и ако тя е true и резулата е true (връща истина, когато поне едната операнда е истина):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| True | || | False | = | True |
| False | || | True | = | True |
| True | || | True | = | True |
| False | || | False | = | False |

4. ^ (xor) - Изключващо или – върща истина, ако двата операнда са различни:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| True | ^ | False | = | True |
| False | ^ | True | = | True |
| True | ^ | True | = | False |
| False | ^ | False | = | False |

II.Побитови оператори – променят битовете на едно число. Прави се „и“, „или“ или “xor” за всеки от битовете:

1. ~ - Тилдата обръща всички битове, т.е. навсякъде, където има 1 го прави 0 и обратното:

* &, | и ^ - са същите, но за всеки бит се разглеждат
* << и >> Побитово изместване с по два бита на ляво или надясно

b<<2 – 00000101 става 00010100

b>>2 – 00000101 става 00000001

III. С Convert.ToString(35,2).PadLeft(32,‘0‘); - може да видим 35 в двойчна бройна система, а ако е 16 на мястото на 2, може да видим 35 в шестнадесетична бройна система. С PadLeft може да кажем колко символа искаме да заема нашият стринг (в случая 32) и ако случайно стрига е по-малко на брой, тогава ислам да ми запълва със символа нула.

IV. Оператори за сравнение:

+=,-=,|=,<<=…

X |= 3 🡪 x = x |3 🡪 тове е равно на x или 3

X <<= 3 🡪 x = x <<3 🡪 тове е равно на x побитово измесено с 3

или обобщено: v1 0= v2; 🡪 v1 = v1 o v2;

* „? :“ - оператора ис: дали даден обект е от даден клас
* ?? – за избягване на дефолтни стойности (пример:вземи еди коя си стойност или 5)

X = null;

Y = x ?? 5;

Cw(y);//резултат 5

V.Инплицитно(автоматично) и експлицитно(изрично) прехвърляне на типове:

- Инплицитно кастване : int i = 5; long l = i;

IV. % - дели с остатък