```
public class UnsignedUtil {
    /**
     * java byte (1 byte == 8 bit) (-2^7~2^7-1 : -128~127) to unsigned short(0~2^8-1 : 0~255)
     *
     * @param data
     * @return
     */
    public static int getUnsignedByte (byte data){
        return data&0x0FF;
    }

    /**
     * java short (1 short == 2 byte == 16 bit) (-2^15~2^15-1 : -32768~32767) to unsigned short(0~2^16-1 : 0~65535)
     *
     * @param data
     * @return
     */
    public static int getUnsignedShort (short data){
        return data&0x0FFFF;
    }

    /**
     * java int (1 int == 4 byte == 32 bit)(-2^31~2^31-1 : -2147483648~2147483647) to unsigned short(0~2^32-1 : 0~4294967295)
     *
     * @param data
     * @return
     */
    public long getUnsignedInt (int data){
        return data&0x0FFFFFFFF;
    }

}
```

java 類型

8bit unsigned integer ———> short

8bit signed integer ---> byte

16bit unsigned integer ---> int

16bit signed integer ---> short

32bit unsigned integer ---> long

32bit signed integer ---> int

java,c,c++ 語言之間基底資料型別的比較 收藏

當要進行底層移植的時候肯定會遇到這些問題。特整理了下。

java 語言基底資料型別

在 JAVA 中一共有八種基底資料型別，他們分別是

byte、short、int、long、float、double、char、boolean

整型

其中 byte、short、int、long 都是表示整數的，只不過他們的取值範圍不一樣

byte 的取值範圍為-128~127，佔用 1 個位元組（-2 的 7 次方到 2 的 7 次方-1）

short 的取值範圍為-32768~32767，佔用 2 個位元組（-2 的 15 次方到 2 的 15 次方-1）

int 的取值範圍為（-2147483648~2147483647），佔用 4 個位元組（-2 的 31 次方到 2 的 31 次方-1）

long 的取值範圍為（-9223372036854774808~9223372036854774807），佔用 8 個位元組（-2 的 63 次方到 2 的 63 次方-1）

可以看到 byte 和 short 的取值範圍比較小，而 long 的取值範圍太大，佔用的空間多，基本上 int 可以滿足我們的日常的計算了，而且 int 也是使用的最多的整型類型了。

在通常情況下，如果 JAVA 中出現了一個整數數位比如 35，那麼這個數字就是 int 型的，如果我們希望它是 byte 型的，可以在資料後加上大寫的 B：35B，表示它是 byte 型的，同樣的 35S 表示 short 型，35L 表示 long 型的，表示 int 我們可以什麼都不用加，但是如果要表示 long 型 的，就一定要在資料後面加“L”。

浮點型

float 和 double 是表示浮點型的資料類型，他們之間的區別在於他們的精確度不同

float 3.402823e+38 ~ 1.401298e-45（e+38 表示是乘以 10 的 38 次方，同樣，e-45 表示乘以 10 的負 45 次方）佔用 4 個位元組

double 1.797693e+308~ 4.9000000e-324 佔用 8 個位元組

double 型比 float 型存儲範圍更大，精度更高，所以通常的浮點型的資料在不聲明的情況下都是 double 型的，如果要表示一個資料是 float 型的，可以在資料後面加上“F”。

浮點型的資料是不能完全精確的，所以有的時候在計算的時候可能會在小數點最後幾位出現浮動，這是正常的。

boolean 型（布林型）

這個類型只有兩個值，true 和 false（真和非真）

boolean t = true；

boolean f = false；

char 型（文本型）

用於存放字元的資料類型，佔用 2 個位元組，採用 unicode 編碼，它的前 128 位元組編碼與 ASCII 相容

字元的存儲範圍在\u0000~\uFFFF，在定義字元型的資料時候要注意加' '，比如 '1'表示字元'1'而不是數值 1，

char c = ' 1 ';

我們試著輸出 c 看看，System.out.println(c);結果就是 1，而如果我們這樣輸出呢 System.out.println(c+0);

結果卻變成了 49。

如果我們這樣定義 c 看看

char c = ' \u0031 ';輸出的結果仍然是 1，這是因為字元'1'對應著 unicode 編碼就是\u0031

char c1 = 'h',c2 = 'e',c3='l',c4='l',c5 = 'o';

System.out.print(c1);System.out.print(c2);System.out.print(c3);System.out.print(c4);Sytem.out.print(c5);

String

在前面我們看到過這樣的定義：

String s = "hello";

System.out.println(s);跟上面的 5 條語句組合起來的效果是一樣的，那麼 String 是個什麼呢？String 是字串，它不是基底資料型別，它是一個類。