Символы

Отдельные символы могут храниться в переменных типа **char**. **Значения** символов указываются в апострофах(**одинарные кавычки**). В примере, создается переменная "**с**" **символьного** типа и присваивается значение '**#**'.

char с = '#';

Чтобы записать в переменную символьного типа значение (**символ**), введенный в консоли, нужно воспользоваться **конвертацией**:

char c = Convert.ToChar(Console.ReadLine());

Представление символов в компьютере

Каждому **символу сопоставляется** некоторый **код** - это некоторое **целое** число. Сам по себе код символа для нас не несет никакой пользы, но важно знать, что в кодовой таблице символы расположены **по порядку**.

То есть код маленькой латинской буквы **b** на единицу больше кода буквы **a** и так далее. Как видно из таблицы, **цифры**, **заглавные** и **строчные** буквы латинского алфавита расположены **по порядку.** Чтобы узнать **код** символа, то есть перевести из типа **char** в тип **int**, нужно написать следующее:

char c = '#';

int code = (int)c;

Console.WriteLine(code); // 35

Можно делать и обратное: **по коду** символа, восстановить **символ**, который соответствует ему по таблице. Например:

int code = 35;

char c = (char)code;

Console.WriteLine(c); // '#’

Console.WriteLine((char)48); // '0'

Сравнение символов

Символы также можно сравнивать. Два символа равны только если **равны** их **коды**. Один символ считается больше другого, если его **код** больше.

char first = 'B';

char second = 'v';

Console.WriteLine(first < second);

Выведется **true**, так как код символа '**B**' меньше кода символа '**v**'. Так как символы латинского алфавита расположены **подряд**,  мы можем проверить **является ли символ буквой латинского алфавита**:

char c = Convert.ToChar(Console.ReadLine());

if (c >= 'A' && c <= 'Z')

{

Console.WriteLine("Заглавная буква английского алфавита");

}

Арифметические операции

Над символами можно совершать математические **операции**. Это будет трактоваться как "возьмем **код** этого символа и сделаем операцию над указанным числом". Так как вместо символов подставляются целые числа, следует, что с символами можно выполнять те же операции, что и с целыми числами. Например:

char c = 'а';

c = (char)(c + 3); // 'd'

char b = '0';

c = (char)(c + b);

c = (char)(c - b);

c = (char)(c \* b);

c = (char)(c / b);

c = (char)(c % b);

Данные операции открывают возможность делать **циклы в виде символов**. Например, выведем **все заглавные буквы** латинского алфавита, ведь их коды идут последовательно:

for (char w = 'A'; w <= 'Z'; w++)

{

Console.WriteLine(w);

}

В переменную цикла **символьного** типа записываются все буквы латинского алфавита, так как каждый раз к символу добавляется число 1, что приводит к тому, что в ней окажутся все буквы.

**Символы можно использовать в любых местах, где допускается использование целого числа.** Например:

char c = 'R';

int x = c - 'A' + 1;

В переменной "**x**" находится "**расстояние**" от заданного символа до **начала** алфавита. Другими словами: **номер** заданной буквы от **начала** алфавита. Здесь **+1** добавлено для **нумерации**, начиная с 1.

Строки

**Строка (string)** - это последовательность **символов**. Чтобы создать строковую переменную нужно:

string s = "Hello, World!";

Чтобы записать в нее значение, введенное в консоли, нужно воспользоваться следующей инструкцией:

s = Console.ReadLine();

Console.WriteLine(s);

Длина строковой переменной(количество символов) определяется с помощью встроенной функции **Length**:

string s = "Hello, World!";

Console.WriteLine(s.Length);

Операции со строками

Над строками определены следующие операции:

1. Операция **сложения**(**+**). Применяется для **сцепления** нескольких строк в одну. Например,

string a = "Hello ";

string b = "world!";

string c = a + b;

Для целых и вещественных чисел, при сложении **не** имеет значения в каком порядке идут операнды. При сложении строк **ИМЕЕТ** **значение расположение** операнд:

string a = "Hello ";

string b = "world!";

string d = b + a;

Итак, сложение работает следующим образом: берется **первый** операнд(**слева** от знака +) и **сцепляется** ко второму операнду(**справа** от знака +), следовательно при изменении операнд местами, результирующая строка получается **разной**.

1. Операции **эквивалентности**(**==, !=**). Данные операции проводят **посимвольное** сравнение двух строк на **равенство**. Результат выполнения операций **всегда** имеет **логический**(**bool**) тип. Строки считаются **равными**, если они совпадают по **длине** и содержат одни и те же **символы** на **соответствующих** местах в строке.

string a = "HELLO";

string b = "Hello";

bool c = a == b;

bool d = a != b;

Хранение строки в компьютере

Пусть есть строка **s** следующего вида:

string s = "Basketball";

Как мы уже знаем, строка - это последовательность символов(**char**), и в памяти оно выглядит следующим образом

Все символы в строке **пронумерованы**. **Нумерация** символов в строке начинается **с нуля**. К отдельным символам строки можно обратиться по номеру (**индексу**) данного символа в строке.  Оператор "**[]**" используется для **доступа** к символу строки **по индексу**. Например:

char a = s[0];

char b = s[9];

Если попробовать обратиться к **несуществующему** индексу, то мы получим **ошибку** во время выполнения программы(индекс находился за пределами строки). Например:

char c = s[10];

char d = s[-1];

char e = s[25];

То есть, в данной строке можно обращаться по индексам от 0 до 9. При других значениях индекса, мы получим **ошибку**.

Не трудно догадаться как вычислить индекс **последнего** символа в строке. Так как индексы начинаются с **нуля**, то **последний** индекс в строке можно найти как  **количество символов в этой строке минус 1**.

string s = "Basketball";

int count = s.Length; // 10

Console.WriteLine("Последний символ равен" + s[count - 1]);

Если считывать символы со строки можно, то **изменять** символы в строке **НЕЛЬЗЯ**! Например:

string s = "Basketball";

s[0] = 'b';

Так как символы в строке **пронумерованы**, мы можем пробежать по строке с помощью **цикла** и обращаться к **каждому** символу. **В переменной цикла** мы будем хранить **индексы** со значениями от **нуля** до **длины** строки **минус один**. Посмотрим пример:

string s = "Beegeek";

int countChar = s.Length;

for (int index = 0; index <= countChar - 1; index++)  
{  
  Console.Write(s[index]);  
}

Теперь мы умеем проходить по строке и получать каждый символ в строке. Давайте посчитаем в строке **количество** заглавных букв 'A':

string s = Console.ReadLine();

int countChar = s.Length;

int countA = 0;

for (int index = 0; index <= countChar - 1; index++)

{  
  if (s[index] == 'A')  
  {  
    countA++;  
  }  
}

Console.WriteLine(countA);