

Преобразования типов

Преобразования типов

- Неявные

```
int i = 0;  
i = 3.541 + 3;           // предупреждение компилятора (?)  
printf("%d", i);         // 6
```

- Явные

```
int i = 0;  
i = (int) 3.541 + 3;  
printf("%d", i);         // 6
```

Неявные преобразования типа

Выполняется когда

- Операнды в арифметическом или логическом выражении имеют разный тип (*обычное арифметическое преобразование*).

```
int i = 3;
```

```
double d = 3.541;
```

```
i + d; // i преобразуется в double (3.0)
```

Обычное арифметическое преобразование

Общие правила:

- Типы всегда приводятся к тому типу, который способен обеспечить наибольший диапазон значений при наибольшей точности.
- Любое арифметическое выражение, включающее в себя целые типы, меньшие чем `int`, перед вычислением всегда преобразуется в `int`.

Обычное арифметическое преобразование

Если хотя бы один из операндов имеет вещественный тип:

long double



double



float

Обычное арифметическое преобразование

Если ни один из операндов не имеет вещественный тип:

...

unsigned long int

↑

long int

↑

unsigned int

↑

int

Неявные преобразования типа

Выполняется когда

- Присваиваются друг другу выражения разных типов (резльтирующим является тип выражения, которому присваивается значение).

```
int i = 3;
```

```
double d = 3.541;
```

```
i = d; // d преобразуется в int (3)
```

Неявные преобразования типа

Выполняется когда

- Передача функции аргумента, тип которого отличается от типа соответствующего формального параметра (тип фактического аргумента приводится к типу параметра).

```
double sqrt(double);  
// целое 2 преобразуется в double 2.0  
printf("%f", sqrt(2));
```


Неявные преобразования типа

Выполняется когда

- Возврат из функции значения, тип которого не совпадает с типом возвращаемого результата, заданным в объявлении (тип фактически возвращаемого значения приводится к объявленному).

```
double sub(int il, int ir)
{
    // результат преобразуется в double
    return il - ir;
}
```

Неявные преобразования типа

(примеры)

// пример 1

```
signed char res, c1 = 100, c2 = 3, c3 = 4;
```

```
res = c1 * c2 / c3;
```

```
printf("%d\n", res);
```

// пример 2.1

```
int si = -1;
```

```
unsigned int ui = 1;
```

```
printf("%d\n", si < ui); // ?
```

Явное преобразование типа

Язык Си предоставляет операцию явного преобразования типа

(тип) выражение;

Операция	Название	Нотация	Класс	Приоритет	Ассоциат.
(<тип>)	Преобразование типа	(<тип>) X	Префиксная	14	Справа налево

Явное преобразование типа

(примеры)

// пример 3

```
double d = 1.234, fract_part;  
fract_part = d - (int) d;
```

// пример 4

```
double mean;  
int sum = 3, count = 2;  
  
mean = sum / count;  
printf("%f\n", mean);    // 1.000  
  
mean = (double) sum / count;  
printf("%f\n", mean);    // 1.500
```

Явное преобразование типа

(примеры)

// пример 5

```
long l;
```

```
int i = 1000;
```

```
l = i * i;           // может случиться переполнение
```

```
l = (long) i * i;
```

```
l = (long) (i * i);  // ошибка
```

Явное преобразование типа

(примеры)

// пример 2.2

```
int si = -1;
```

```
unsigned int ui = 1;
```

// ui = 1 представимо в типе int

```
printf("%d\n", si < (int)ui);
```

// пример 2.3

```
int si = /*какое-то число со знаком*/;
```

```
unsigned int ui = /*какое-то число без знака*/;
```

```
printf("%d\n", (si < 0 || (unsigned) si < ui));
```

Литература

1. С. Прата «Язык программирования Си» (глава 5 («Преобразование типов», «Операция приведения»))
2. Б. Керниган, Д. Ритчи «Язык программирования Си» (раздел 2.7)
3. Черновик стандарта C99