

Лабораторная работа № 3.

Тема: Приведение типов. Метод Parse(), класс Convert.

Базовые типы C#:

Тип	Размер в байтах	Тип .NET	Описание
Базовый тип			
object		Object	Может хранить все что угодно, т.к. является всеобщим предком
Логический тип			
bool	1	Boolean	true или false
Целые типы			
sbyte	1	SByte	Целое со знаком (от -128 до 127)
byte	1	Byte	Целое без знака (от 0 до 255)
short	2	Int16	Целое со знаком (от -32768 до 32767)
ushort	2	UInt16	Целое без знака (от 0 до 65535)
int	4	Int32	Целое со знаком (от -2147483648 до 2147483647)
uint	4	UInt	Целое число без знака (от 0 до 4 294 967 295)
long	8	Int64	Целое со знаком (от -9223372036854775808 до 9223372036854775807)
ulong	8	UInt64	Целое без знака (от 0 до 0xffffffffffffffff)
Вещественные типы			
float	4	Single	Число с плавающей точкой двойной точности. Содержит значения приблизительно от $-1.5 \cdot 10^{-45}$ до $+3.4 \cdot 10^{38}$ с 7 значащими цифрами
double	8	Double	Число с плавающей точкой двойной точности. Содержит значения приблизительно от $-5.0 \cdot 10^{-324}$ до $-1.7 \cdot 10^{308}$ с 15-16 значащими цифрами
Символьный тип			
char	2	Char	Символы Unicode
Строковый тип			
string		String	Строка из Unicode-символов
Финансовый тип			
decimal	12	Decimal	Число до 28 знаков с фиксированным положением десятичной точки. Обычно используется в финансовых расчетах. Требует суффикса <<m>> или <<M>>

Задание:

1. Преобразование строк в числа с помощью Parse().

Напишите программу, которая запрашивает у пользователя три строки, содержащие целые числа. Преобразуйте каждую строку в число с помощью метода int.Parse() и найдите их сумму. Подсказка: Используйте: int number = int.Parse(input);

Пример:

```
Введите первое число: 10
Введите второе число: 20
Введите третье число: 30
Сумма: 60
```

2. Использование TryParse() для обработки ошибок.

Напишите программу, которая запрашивает у пользователя строку и пытается преобразовать её в число с помощью int.TryParse(). Если преобразование успешно, выведите число, иначе сообщите об ошибке. Подсказка: bool success = int.TryParse(input, out int result);

Пример:

```
Введите число: 123
Преобразованное число: 123

Введите число: abc
Ошибка: Введено некорректное число.
```

3. Преобразование разных типов через класс Convert.

Напишите программу, которая запрашивает у пользователя строку, содержащую число с плавающей запятой. Используйте класс Convert для преобразования строки в число типа double и выведите результат, округлив его до целого числа.

Подсказка:

```
double value = Convert.ToDouble(input);
```

```
int rounded = (int)Math.Round(value);
```

Пример:

```
Введите число с плавающей запятой: 12,34
Округленное число: 12
```

Задание 4. Преобразование типов с потерей данных.

Напишите программу, которая запрашивает у пользователя число типа double и преобразует его в int. Покажите, что дробная часть теряется. Подсказка: int

```
intValue = (int)doubleValue;
```

Пример:

```
Введите число с плавающей запятой: 9,99
Результат после преобразования в int: 9
```

5. Преобразование типов между bool, int и string.

Создайте программу, которая запрашивает у пользователя строку. Если пользователь ввел "true" или "false", преобразуйте это в bool и затем в int (true -> 1, false -> 0). В остальных случаях сообщите об ошибке.

Подсказка:

```
bool success = bool.TryParse(input, out bool boolValue);
```

```
int intValue = Convert.ToInt32(boolValue);
```

Пример:

```
Введите логическое значение (true/false): true
Преобразованное значение в int: 1

Введите логическое значение (true/false): abc
Ошибка: Некорректное логическое значение.
```

6. Преобразование Unicode-значения в символ.

Напишите программу, которая запрашивает у пользователя целое число от 32 до 126 и преобразует его в символ Unicode с помощью Convert.

Подсказка: char symbol = Convert.ToChar(number);

Пример:

```
Введите число: 65
Соответствующий символ: A
```

7. Преобразование времени с использованием Convert.

Попросите пользователя ввести количество часов и минут как строки.

Преобразуйте их в числа и рассчитайте общее количество минут.

Подсказка:

```
int hours = Convert.ToInt32(hoursInput);
```

```
int minutes = Convert.ToInt32(minutesInput);
```

Пример:

```
Введите количество часов: 2
Введите количество минут: 30
Общее количество минут: 150
```

8. Преобразование строки с валютой в число.

Создайте программу, которая запрашивает у пользователя строку с суммой в рублях (например, "123 руб."). Удалите символы, оставив только число, и преобразуйте его в int.

Подсказка:

```
string input = "123 руб.".Replace(" руб.", "");
int amount = int.Parse(input);
```

Пример:

```
Введите сумму: 123 руб.
Преобразованная сумма: 123
```

9. Преобразование температуры.

Запросите у пользователя температуру в градусах Цельсия и преобразуйте её в градусы Фаренгейта. Формула:

$$F = \frac{9}{5} \times C + 32$$

Подсказка: Используйте Convert.ToDouble() для работы с дробными значениями.

Пример:

```
Введите температуру в Цельсиях: 25
Температура в Фаренгейтах: 77
```

10. Проверка на корректный ввод целого числа.

Напишите программу, которая запрашивает у пользователя строку и проверяет, можно ли преобразовать её в int с помощью int.TryParse(). Если ввод корректен, увеличьте число на 10 и выведите результат. В противном случае сообщите об ошибке.

Пример:

```
Введите число: 50
Результат: 60

Введите число: abc
Ошибка: Введено некорректное число.
```

11. Преобразование строки в дату с помощью DateTime.Parse()

Создайте программу, которая запрашивает у пользователя дату в формате "dd.ММ.yyyy" и преобразует её в объект DateTime. Затем выведите эту дату в формате "MMMM dd, yyyy" (например, "October 16, 2024").

Подсказка:

```
DateTime date = DateTime.Parse(input);
Console.WriteLine(date.ToString("MMMM dd, yyyy"));
```

Пример:

```
Введите дату (dd.ММ.yyyy): 16.10.2024
Дата в новом формате: October 16, 2024
```

12. Преобразование чисел из двоичной строки

Напишите программу, которая запрашивает у пользователя строку, содержащую двоичное число (например, "1011"). Преобразуйте её в целое число с помощью Convert.ToInt32() и выведите результат.

Подсказка:

```
int number = Convert.ToInt32(input, 2);
```

Пример:

```
Введите двоичное число: 1011
Целое число: 11
```

13. Преобразование чисел в шестнадцатеричную систему

Создайте программу, которая запрашивает у пользователя целое число и преобразует его в шестнадцатеричную строку.

Подсказка:

```
string hex = Convert.ToString(number, 16).ToUpper();
```

Пример:

```
Введите число: 255
Шестнадцатеричное представление: FF
```

14. Сумма чисел из строки с разделителями

Запросите у пользователя строку, содержащую числа, разделённые запятыми (например, "10,20,30"). Преобразуйте каждое число в int и выведите их сумму.

Подсказка:

```
string[] parts = input.Split(',');
int sum = parts.Sum(int.Parse);
```

Пример:

```
Введите числа через запятую: 10,20,30
Сумма чисел: 60
```

15. Преобразование символов в ASCII-коды

Напишите программу, которая запрашивает у пользователя строку и выводит ASCII-коды для каждого символа в строке.

Подсказка:

```
foreach (char c in input)
{
    Console.WriteLine($"{c}: {(int)c}");
}
```

Пример:

```
Введите строку: Hi
H: 72
i: 105
```

16. Преобразование градусов в радианы и наоборот

Создайте программу, которая запрашивает у пользователя угол в градусах и преобразует его в радианы. Затем запрашивает у пользователя угол в радианах и переводит в градусы и выводит результаты.

Подсказка: Используйте Math.PI для значения π.

Формулы:

$$\text{Радианы} = \text{Градусы} \times \frac{\pi}{180}$$

$$\text{Градусы} = \text{Радианы} \times \frac{180}{\pi}$$

Пример:

```
Введите угол в градусах: 180
Угол в радианах: 3,14159
```

```
Введите угол в радианах: 7
Угол в градусах: 401,2738
```

17. Преобразование между типами decimal и double.

Напишите программу, которая запрашивает у пользователя дробное число и преобразует его в decimal и double. Покажите, что при преобразовании в double возможна потеря точности.

Подсказка:

```
double dValue = Convert.ToDouble(input);
decimal decValue = Convert.ToDecimal(input);
```

Пример:

```
Введите число: 1.234567890123456789
double: 1,23456789012346
decimal: 1,234567890123456789
```

18. Проверка на переполнение при приведении типов.

Напишите программу, которая запрашивает у пользователя число типа long и пытается привести его к типу int. Если значение выходит за пределы int, обработайте ошибку с помощью исключения.

Подсказка: Используйте try-catch для обработки переполнения:

```
try
{
    int result = checked((int)longValue);
}
catch (OverflowException)
{
    Console.WriteLine("Ошибка: Значение выходит за пределы int.");
}
```

Пример:

```
Введите большое число: 2147483648
Ошибка: Значение выходит за пределы int.
```

19. Преобразование объекта в строку и обратно.

Создайте программу, которая запрашивает у пользователя целое число и преобразует его в строку с использованием метода ToString(). Затем используйте int.Parse() для преобразования строки обратно в число и проверьте, равны ли исходное и преобразованное значения.

Пример:

```
Введите число: 42
Преобразованное значение: 42
Значения равны? True
```

20. Преобразование времени в секунды и обратно

Напишите программу, которая запрашивает у пользователя часы, минуты и секунды. Преобразуйте это время в общее количество секунд. Затем преобразуйте секунды обратно в часы, минуты и секунды.

Подсказка:

```
int totalSeconds = hours * 3600 + minutes * 60 + seconds;
int h = totalSeconds / 3600;
int m = (totalSeconds % 3600) / 60;
int s = totalSeconds % 60;
```

Пример:

```
Введите часы: 1
Введите минуты: 30
Введите секунды: 45
Общее количество секунд: 5445
Обратно: 1 час(ов), 30 минут, 45 секунд
```