

■ Podstawy C#

Podstawowe typy danych w C#:

- `int` (liczba całkowita);
- `double` (liczba rzeczywista podwójnej precyzji);
- `string` (łańcuch tekstowy);
- `bool` (wartość logiczna – true, false);
- `decimal` (liczba rzeczywista – typ używany do danych walutowych).
- `char` (pojedynczy znak)

Typ `string` należy do typów referencyjnych (ang. Reference Type), pozostałe należą do typów wartościowych (inna nazwa: proste lub skalarne, ang. Value Type)

```
static void p11()
{
    Console.WriteLine("Hello World!!!");
}
```

```
static void p12()
{
    Console.Write("Podaj promień: ");
    double r = Double.Parse(Console.ReadLine());
    double obwod = 2 * Math.PI * r;
    Console.WriteLine("Obwód wynosi: {0:0.##}", obwod);
}
```

Wczytywanie danych ze strumienia wejściowego

Metoda `Console.ReadLine()` zwraca zmienną typu łańcuchowego (`string`). Aby wczytane dane można było traktować jako zmienne typu `int`, `double`, itd., konieczne jest konwersja z typu `string` na typ `int`, `double`. Do konwersji można wykorzystać statyczne metody klasy `Convert`, np. `Convert.ToInt32(...)`, `Convert.ToDouble(...)` lub metodę `Parse`, np. `Int32.Parse(...)`, `Double.Parse(...)`.

Wypisywanie danych do strumienia wyjściowego

Do wypisywania danych używa się metody `Console.WriteLine(...)`, lub `Console.Write(...)`. Różnica pomiędzy tymi metodami polega jedynie na tym, że w pierwszym przypadku do strumienia wyjściowego dopisany zostanie znak końca nowej linii.

Formatowanie danych w metodzie `Console.WriteLine()`

```
{Index[:FormatItem]}
```

Index oznacza, która zmienna podlega formatowaniu. Numerowanie rozpoczyna się od zera. *FormatItem* może przyjmować jedną z predefiniowanych wartości (`C` – waluta, `N` – liczba, `X` – kod szesnastkowy, `E` – naukowy, `F` – stałopozycyjny). Przykładowy format może wyglądać następująco: `{0:C}`, co oznacza, że liczba na pozycji pierwszej sformatowana zostanie jak waluta, tj. na końcu dopisany zostanie symbol waluty. Możliwe jest też podanie explicite formatu, np.: `{0:0.00}` – liczba na pozycji pierwszej będzie wypisana z dwoma miejscami po przecinku (jeśli jest więcej cyfr niż dwie – pozostałe zostaną obcięte, jeśli mniej – zostanie dopisana odpowiednia liczba zer), `{1:0.##}` – liczba na pozycji drugiej będzie wypisana z co najwyżej dwoma miejscami po przecinku (jeśli jest więcej cyfr niż dwie – zostaną obcięte, jeśli mniej – wypisane zostanie tylko tyle ile jest).

Instrukcja warunkowa

```
if ( warunek )
{
    BlokInstrukcji_1
}
[else
{
    BlokInstrukcji_2
}]
```

Zadanie 1.1: Napisać program do obliczania pola koła o zadanym (wczytanym przez użytkownika) promieniu. Jeśli użytkownik wprowadzi liczbę ujemną, program powinien wypisać komunikat „Niepoprawne dane”.

Zadanie 1.2: Napisać program, który wczytuje dwie liczby (długości boków pewnego prostokąta). Program powinien obliczyć pole prostokąta i sprawdzić, czy prostokąt może być kwadratem.

Zadanie 1.3: Napisać program, który sprawdza, czy wczytana liczba całkowita jest parzysta.

Zadanie 1.4: Napisać program wczytujący 3 liczby i znajdujący największą z nich.

Zadanie 1.5: Napisać program do znajdowania miejsc zerowych równania kwadratowego. Należy rozważyć trzy przypadki: delta większa/mniejsza/równa zero. Dla przypadku $\Delta < 0$ obliczyć oddzielnie część rzeczywistą i część urojoną i wypisać na ekranie w sposób jaki zapisujemy liczby zespolone, np. $2 + 3i$.

Instrukcja switch

```
switch ( wyrażenie )
{
    case przypadek1:
        BlokInstrukcji_1
        break;
    [default:
        BlokInstrukcji_2
        break;]
}
```

```
static void p13()
{
    Console.WriteLine("Podaj liczbę całkowitą z przedziału [2 4]: ");
    int a = Int32.Parse(Console.ReadLine());

    switch (a)
    {
        case 2:
            Console.WriteLine("Podajesz liczbę 2");
            break;

        case 3:
            Console.WriteLine("Podajesz liczbę 3");
            break;

        case 4:
            Console.WriteLine("Podajesz liczbę 4");
            break;

        default:
            Console.WriteLine("Podajesz liczbę spoza przedziału [2
                                4]");
            break;
    }
}
```

Zadanie 1.6: Napisać program do przeliczania podanej przez użytkownika kwoty (w zł) na określoną walutę (również podaną przez użytkownika).