

ChatGPT

Wprowadzanie danych

W języku C# możesz wprowadzać dane od użytkownika za pomocą klasy Console.

Oto kilka przykładów wprowadzania różnych typów danych:

Wprowadzanie ciągu znaków (string):

```
Console.Write("Podaj imię: ");  
  
string imie = Console.ReadLine();  
  
Console.WriteLine("Witaj, " + imie + "!");
```

Wprowadzanie liczby całkowitej (int):

```
Console.Write("Podaj liczbę całkowitą: ");  
  
int liczbaCalkowita;  
  
while (!int.TryParse(Console.ReadLine(), out liczbaCalkowita))  
{  
    Console.Write("To nie jest poprawna liczba całkowita. Spróbuj ponownie: ");  
}  
  
Console.WriteLine("Wprowadziłeś liczbę całkowitą: " + liczbaCalkowita);
```

Wprowadzanie liczby zmiennoprzecinkowej (double):

```
Console.Write("Podaj liczbę zmiennoprzecinkową: ");  
  
double liczbaZmiennoprzecinkowa;  
  
while (!double.TryParse(Console.ReadLine(), out liczbaZmiennoprzecinkowa))  
{  
    Console.Write("To nie jest poprawna liczba zmiennoprzecinkowa. Spróbuj ponownie: ");  
}
```

```
Console.WriteLine("Wprowadziłeś liczbę zmiennoprzecinkową: " + liczbaZmiennoprzecinkowa);
```

Wprowadzanie pojedynczego znaku (char):

```
Console.Write("Podaj pojedynczy znak: ");
char pojedynczyZnak;
while (!char.TryParse(Console.ReadLine(), out pojedynczyZnak) || Console.ReadLine().Length != 1)
{
    Console.Write("To nie jest pojedynczy znak. Spróbuj ponownie: ");
}
Console.WriteLine("Wprowadziłeś znak: " + pojedynczyZnak);
```

Powyższe przykłady używają `Console.ReadLine()` do wczytywania danych w formie tekstu. Następnie są one konwertowane na odpowiedni typ (jeśli to możliwe) za pomocą metod takich jak `int.TryParse`, `double.TryParse`, itp. Warto zauważyć, że `TryParse` jest używane, aby uniknąć błędów, gdy użytkownik wprowadzi nieprawidłowe dane.

Oto przykład prostego programu w C#, który **pobiera od użytkownika ciąg znaków i następnie wypisuje go na ekranie:**

```
using System;

class Program
{
    static void Main()
    {
        // Wprowadzanie ciągu znaków
        Console.Write("Podaj ciąg znaków: ");
        string ciagZnakow = Console.ReadLine();

        // Wyświetlanie wprowadzonego ciągu
        Console.WriteLine("Wprowadziłeś: " + ciagZnakow);

        // Oczekiwanie na naciśnięcie klawisza przed zamknięciem programu
        Console.ReadKey();
    }
}
```

Oto przykład prostego programu w C#, który pobiera od użytkownika **liczbę całkowitą i następnie wypisuje ją na ekranie:**

```
using System;

class Program
{
```

```

static void Main()
{
    // Wprowadzanie liczby całkowitej
    Console.Write("Podaj liczbę całkowitą: ");

    // Używamy int.TryParse do bezpiecznego przekształcenia wprowadzonego tekstu na liczbę
    całkowitą
    if (int.TryParse(Console.ReadLine(), out int wprowadzonaLiczba))
    {
        // Wyświetlanie wprowadzonej liczby
        Console.WriteLine("Wprowadziłeś liczbę całkowitą: " + wprowadzonaLiczba);
    }
    else
    {
        // Obsługa przypadku, gdy wprowadzony tekst nie jest liczbą całkowitą
        Console.WriteLine("To nie jest poprawna liczba całkowita.");
    }
}

```

Poniżej znajdziesz program w C#, który **pobiera dwie liczby całkowite od użytkownika**, a następnie dodaje je i zapisuje wynik w nowej zmiennej:

```

using System;

class Program
{
    static void Main()
    {
        // Wprowadzanie pierwszej liczby całkowitej
        Console.Write("Podaj pierwszą liczbę całkowitą: ");

        // Używamy int.TryParse do bezpiecznego przekształcenia wprowadzonego tekstu na liczbę
        całkowitą
        if (int.TryParse(Console.ReadLine(), out int pierwszaLiczba))
        {
            // Wprowadzanie drugiej liczby całkowitej
            Console.Write("Podaj drugą liczbę całkowitą: ");

            // Używamy int.TryParse do bezpiecznego przekształcenia wprowadzonego tekstu na liczbę
            całkowitą
            if (int.TryParse(Console.ReadLine(), out int drugaLiczba))
            {
                // Dodawanie liczb i zapis wyniku w nowej zmiennej
                int suma = pierwszaLiczba + drugaLiczba;

                // Wyświetlanie wyniku
                Console.WriteLine("Suma liczb {pierwszaLiczba} i {drugaLiczba} wynosi: {suma}");
            }
            else
            {
                Console.WriteLine("To nie jest poprawna druga liczba całkowita.");
            }
        }
    }
}

```

```

    }
}
else
{
    Console.WriteLine("To nie jest poprawna pierwsza liczba całkowita.");
}

// Oczekiwanie na naciśnięcie klawisza przed zamknięciem programu
Console.ReadKey();
}
}

// Oczekiwanie na naciśnięcie klawisza przed zamknięciem programu
Console.ReadKey();
}
}

```

W tym programie, użytkownik wprowadza dwie liczby całkowite, a następnie program próbuje przekształcić te wprowadzone dane na liczby całkowite za pomocą `int.TryParse()`. Jeśli obie konwersje się powiedą, program dodaje te liczby i wyświetla wynik na ekranie. Jeśli konwersje nie powiedą się, program informuje użytkownika o błędzie.

Prosty program okienkowy w C#

Aby stworzyć prosty okienkowy program w C#, można użyć platformy Windows Forms, która jest często stosowana do tworzenia interfejsów użytkownika w aplikacjach desktopowych. Poniżej znajduje się prosty przykład programu, który tworzy okno z przyciskiem i obsługą zdarzenia kliknięcia tego przycisku:

```

// Prosty program okienkowy C#

using System;
using System.Windows.Forms;

class OknoProgramu : Form
{
    private Button przycisk;

    public OknoProgramu()
    {
        // Ustawienia okna
        Text = "Prosty Program";
    }
}

```

```
Size = new System.Drawing.Size(300, 200);

// Tworzenie przycisku
przycisk = new Button();
przycisk.Text = "Kliknij mnie!";
przycisk.Location = new System.Drawing.Point(50, 50);

// Dodawanie obsługi zdarzenia kliknięcia przycisku
przycisk.Click += Przycisk_Click;

// Dodawanie przycisku do okna
Controls.Add(przycisk);
}

private void Przycisk_Click(object sender, EventArgs e)
{
    MessageBox.Show("Witaj, to jest prosty program w C#!");
}

[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.Run(new OknoProgramu());
}
}
```

Ten program tworzy prosty interfejs z jednym oknem i jednym przyciskiem. Po kliknięciu przycisku wyświetla się komunikat powitalny.

Powyższy kod musi być uruchomiony w projekcie obsługującym Windows Forms. Można utworzyć taki projekt w Visual Studio, wybierając nowy projekt typu "Windows Forms App".