Operatory arytmetyczne i logiczne w C#

Wprowadzenie

Operatory to specjalne symbole, które pozwalają na wykonywanie różnych operacji na danych. W programowaniu dzielimy je na kilka kategorii, ale w tej dokumentacji skupimy się na dwóch najważniejszych: **operatorach arytmetycznych** i **operatorach logicznych**. Poznasz ich działanie oraz przykłady użycia w Unity.

Operatory arytmetyczne

Operatory arytmetyczne służą do wykonywania podstawowych operacji matematycznych, takich jak dodawanie, odejmowanie, mnożenie czy dzielenie. Oto najważniejsze z nich:

1. Dodawanie (`+`)

- Dodaje dwie liczby lub łączy dwa ciągi znaków.

```
int a = 5;
int b = 3;
int wynik = a + b; // wynik = 8
```

2. Odejmowanie (`-`)

- Odejmuje jedną liczbę od drugiej.

```
int a = 10;
int b = 4;
int wynik = a - b; // wynik = 6
```

3. Mnożenie (`*`)

- Mnoży dwie liczby.

```
int a = 7;
int b = 6;
int wynik = a * b; // wynik = 42
```

4. Dzielenie (`/`)

- Dzieli jedną liczbę przez drugą. Uwaga: Dzielenie liczb całkowitych daje wynik całkowity!

```
int a = 10;
int b = 3;
int wynik = a / b; // wynik = 3 (bo 10 / 3 = 3.333, ale wynik jest typu int)
```

5. Reszta z dzielenia (`%`)

- Zwraca resztę z dzielenia jednej liczby przez drugą.

```
int a = 10;
int b = 3;
int wynik = a % b; // wynik = 1 (bo 10 / 3 = 3 z resztą 1)
```

6. Inkrementacja (`++`) i dekrementacja (`--`)

- Zwiększa lub zmniejsza wartość zmiennej o 1.

```
int a = 5;
a++; // a = 6
a--; // a = 5
```

Operatory logiczne

Operatory logiczne służą do porównywania wartości i tworzenia warunków. Są używane głównie w instrukcjach warunkowych (`if`, `while`, itp.).

1. Równość (`==`)

- Sprawdza, czy dwie wartości są równe.

```
int a = 5;
int b = 5;
bool wynik = (a == b); // wynik = true
```

2. Nierówność (`!=`)

- Sprawdza, czy dwie wartości są różne.

```
int a = 5;
int b = 3;
bool wynik = (a != b); // wynik = true
```

3. Większe niż (`>`)

- Sprawdza, czy jedna wartość jest większa od drugiej.

```
int a = 10;
int b = 5;
bool wynik = (a > b); // wynik = true
```

4. Mniejsze niż (`<`)

- Sprawdza, czy jedna wartość jest mniejsza od drugiej.

```
int a = 3;
int b = 7;
bool wynik = (a < b); // wynik = true
```

5. Większe lub równe (`>=`)

- Sprawdza, czy jedna wartość jest większa lub równa drugiej.

```
int a = 10;
int b = 10;
bool wynik = (a >= b); // wynik = true
```

6. Mniejsze lub równe (`<=`)

- Sprawdza, czy jedna wartość jest mniejsza lub równa drugiej.

```
int a = 5;
int b = 10;
bool wynik = (a <= b); // wynik = true
```

7. Logiczne AND (`&&`)

- Zwraca `true`, jeśli oba warunki są prawdziwe.

```
bool warunek1 = true;
bool warunek2 = false;
bool wynik = (warunek1 && warunek2); // wynik = false
```

8. Logiczne OR (`||`)

- Zwraca `true`, jeśli przynajmniej jeden z warunków jest prawdziwy.

```
bool warunek1 = true;
bool warunek2 = false;
bool wynik = (warunek1 || warunek2); // wynik = true
```

9. Logiczne NOT (`!`)

- Neguje wartość logiczną (zamienia `true` na `false` i odwrotnie).

```
bool warunek = true;
bool wynik = !warunek; // wynik = false
```

Przykład praktyczny w Unity

Użycie operatorów w skrypcie:

```
using UnityEngine;

public class OperatoryPrzyklad : MonoBehaviour
{
    void Start()
    {
        // Operatory arytmetyczne
        int zdrowie = 100;
        int obrazenia = 20;
        int noweZdrowie = zdrowie - obrazenia; // zdrowie = 80
        Debug.Log("Nowe zdrowie: " + noweZdrowie);

        // Operatory logiczne
```

```
        bool czyGraczJestZywy = (noweZdrowie > 0);
        bool czyGraczMaPelneZdrowie = (noweZdrowie == 100);

        if (czyGraczJestZywy && !czyGraczMaPelneZdrowie)
        {
            Debug.Log("Gracz żyje, ale nie ma pełnego zdrowia.");
        }
    }
}
```

Wyjaśnienie:

1. **Operatory arytmetyczne**:
    - Obliczamy nowe zdrowie gracza po otrzymaniu obrażeń.
2. **Operatory logiczne**:
    - Sprawdzamy, czy gracz żyje i czy ma pełne zdrowie.
    - Używamy operatora `&&` (AND) oraz `!` (NOT), aby określić warunek.

Podsumowanie

Operatory arytmetyczne:

- Służą do wykonywania operacji matematycznych.
- Przykłady: `+`, `-`, `*`, `/`, `%`, `++`, `--`.

Operatory logiczne:

- Służą do porównywania wartości i tworzenia warunków.
- Przykłady: `==`, `!=`, `>`, `<`, `>=`, `<=`, `&&`, `||`, `!`.

Zrozumienie operatorów jest kluczowe do pisania efektywnego kodu w Unity. Dzięki nim możemy wykonywać obliczenia, porównywać wartości i kontrolować przepływ programu.