

Lab 1

Zagadnienia do opanowania:

- metoda `Console.WriteLine`,
 - tworzenie projektów wraz z bibliotekami dll,
 - tablice
-

Temat: Wprowadzenie

1. Utworzyć projekt Minutnik jako *Console Application*.
2. Zaimplementować zegar odliczający czas do końca zajęć i wyświetlający go w prawym górnym rogu okna aplikacji.

Podpowiedzi:

- struktura przeznaczona do obsługi czasu: `System.DateTime`
- aktualny czas: `System.DateTime.Now`
- ustawienie pozycji kursora: `Console.SetCursorPosition()`
- użyć formatowania łańcuchów

Temat: Biblioteka DLL

3. Na podstawie Minutnika, wydzielić klasę Minutnik z funkcjonalnością pozwalającą w konstruktorze podać czas końcowy i metodę pozwalającą pobrać zarówno sformatowany czas do końca jak i jego składowe (godziny, minuty i sekundy) oraz cały czas jako `TimeSpan`

Temat: Tablice

4. Utworzyć tablicę 10e6 elementów typu `int` i wstawić do tej tablicy te elementy.
5. Wyliczyć sumę elementów w tablicy.
6. Dla punktów 4 i 5 zmierzyć czas trwania tych operacji (wykorzystać `DateTime.Now`).
7. Powtórzyć punkty 4, 5 i 6 dla tablicy obiektów (`object[]`), do której wstawiamy obiekty typu `int`.

Temat: Parametry metod

8. Zaimplementować metodę `Sum` która wyliczy sumę dowolnej liczby argumentów (`params`).
`public int Sum(params int[] arg);`
9. Zaimplementować metodę `SumAvg`, która wyliczy sumę dowolnej liczby argumentów oraz ich średnią.
`public int SumAvg(out float avg, params int[] arg);`
10. (wartości domyślne argumentów) Zaimplementować metodę `Div` w następujący sposób
`public float Div(float x, float y = 5)`
11. (argumenty nazwane) Wywołać metodę `Div` podając wartości argumentów w odwrotnej kolejności niż w zad 10.

Pytania kontrolne:

- Jaka jest różnica między modyfikatorem `ref` i `out`? (przypisanie wartości wewnątrz metody i wartość początkowa parametru)
- Czy jedna metoda może mieć dwa parametry z modyfikatorem `params`?
- Czy argument z modyfikatorem `params` może być pierwszy? Innymi słowy, czy poprawna jest definicja:
`public int SumDiv(params int[] arg, ref int x, out float res)`
- Czy używając nazw argumentów należy podać je dla wszystkich używanych argumentów?
- Czy argumenty z parametrami `out/ref` mogą mieć wartości domyślne?

Temat: typy wartościowe i referencyjne

13. Zdefiniować strukturę `Point` o dwóch polach `X` i `Y` typu `int` oraz metodę `Print()`, która wyświetla wartości tych pól.

14. Sprawdzić, które wywołania są poprawne. Dlaczego?

```
Point p1;           Point p2;           Point p3 = new Point();
p1.X = 1;           p2.Print();          p3.Print();
p1.Y = 1;
p1.Print();
```

15. Utworzyć zmienną `p1` typu `Point` i nadać wartości 1,1 jej polom. Następnie utworzyć zmienną `p2` tego samego typu w następujący sposób:
`Point p2 = p1;`
jakie wartości zostaną wyświetlone po wywołaniu metody `Print` dla zmiennej `p2`?
16. Kontynuując powyższy przykład, jeżeli zmienimy wartość pola `X` zmiennej `p1`, co wyświetlą metody `Print` dla pól `p1` i `p2`?
17. Powtórzyć zadania 14, 15 i 16 zmieniając typ `Point` ze struktury na klasę.
18. Napisać dowolną metodę której argumentem jest typ `Point` (najpierw jako struktura, a potem jako klasa). Czy zmiany dokonane na zmiennej będącej argumentem tej metody są widoczne poza jej ciałem?

Pytania kontrolne:

- Co się stanie w zad 15, jeśli struktura `Point` będzie miała dodatkowo pole będące obiektem jakiejś klasy?
- Czy można przekazać typ referencyjny przez wartość i referencję?