

Laboratorium 4

1. Ćwiczenie 1 Przekazywanie parametrów do funkcji

1. W programie zdefiniuj metody:

```
void Fun1(in int i)
{
    ..
}

void Fun2(out int i)
{
    ..
}

void Fun3(ref int i)
{
    ..
}

void Fun4(int i)
{
    ..
}
```

wewnątrz których zostaną zmodyfikowane wejściowe zmienne, będące parametrami funkcji.

Spróbuj:

- Zmodyfikować wejściowe zmienne wewnątrz funkcji.
- Przekazać do funkcji zmienną typu **int**.
- Zdefiniować zmienną typu **short** i spróbować przekazać ją do funkcji.
- Zdefiniować wewnątrz klasy dwie metody o tej samej nazwie, różniące się jedynie modyfikatorem **in** i **out**.
- Wypisz wartości x i y przed i po wywołaniu funkcji.

2. Stwórz obiekt **Point**.

```
public class Point
{
    public int x, y;

    public Point(int p1, int p2)
    {
        x = p1;
        y = p2;
    }
}
```

Przełącz go odpowiednio do metody Fun5 i Fun6:

```
void Fun5(Point p)
{
    ..
}
```

```
void Fun6(ref Point p)
{
    ..
}
```

Przełącz do obu funkcji obiekt `Point`.

- Wewnątrz funkcji utwórz inny obiekt `Point`, zainicjuj `x` i `y` innymi wartościami i przypisz do zmiennej będącej parametrem funkcji utworzony obiekt.
- Wypisz wartości `x` i `y` przed i po wywołaniu funkcji.
- Przypisz do obiektu wskaźnik `null`.
- Spróbuj wypisać `x` i `y` przed i po wywołaniu funkcji.

2. Ćwiczenie 2 Bezpośredni dostęp do pamięci

Uwaga, może wymagać kompilacji z flagą `/unsafe` (np. przez ustawienie w opcjach kompilatora).

- Operacje bezpośrednio na pamięci. Napisz i skompiluj następujący fragment:

```
class TestPointer {
    public unsafe static void Main() {
        int[] list = {10, 100, 200};
        fixed(int *ptr = list)

        for (int i = 0; i < 3; i++) {
            Console.WriteLine("Adres [{0}]={1}", i, (int)(ptr + i));
            Console.WriteLine("Wartość [{0}]={1}", i, *(ptr + i));
        }

        Console.ReadKey();
    }
}
```

Co się stanie jeśli usuniemy modyfikator `unsafe`?

- Poniższą funkcję:

```
public unsafe void swap(int* p, int *q)
{
    int temp = *p;
    *p = *q;
    *q = temp;
}
```

Wywołaj odpowiednio w programie i wypisz efekt jej działania.

- Spróbuj załokować bufor pamięci typu `int` o rozmiarze 1024. Wykonaj na nim bezpośrednie operacje (przypisanie wartości i wypisanie wartości) stosując konstrukcję `*(buf + i)`. Zastosuj odpowiednio słowa kluczowe `unsafe` oraz `fixed`.

3. Ćwiczenie 3 Tablice I

Napisz program, który pozwoli na wstawienie elementu podanego z klawiatury do tablicy jednowymiarowej (pozostałe są przesuwane na koniec). Rozmiar tablicy 1x10.

4. Ćwiczenie 4 Tablice II

Napisz program, który wczytuje 5 liczb z klawiatury do tablicy jednowymiarowej, po czym wyświetla je w odwrotnej kolejności.

5. Ćwiczenie 5 Tablice III

Napisz program, który dla wypełnionej tablicy jednowymiarowej zlicza ilość powtarzających się liczb. Tablica 1x5.

6. Ćwiczenie 6 Tablice IV

Napisz program, który dla sumuje dwie tablice i wynik umieszcza w trzeciej. Tablica ma rozmiar 5x5.

Wypisz własności tablicy wynikowej: Length, LongLength, Rank.

7. Ćwiczenie 7 Tablice V

Napisz program wyliczający wyznacznik tablicy 3 x 3, dla następujących wartości w tablicy:

[0], [0]: 1

[0], [1]: 0

[0], [2]: -1

[1], [0]: 0

[1], [1]: 0

[1], [2]: 1

[2], [0]: -1

[2], [1]: -1

[2], [2]: 0

8. Ćwiczenie 8 Tablice VI

Utwórz następującą tablicę schodkową, zainicjuj podanymi wartościami i wypisz w konsoli:

1	2	3			
4	5	6	7	8	9
10	11	12	13		
14	15	16	17	18	
19	20	21			

Spróbuj zainicjalizować tablicę schodkową na dwa sposoby.