

תרגיל 1

את הקוד יש להגיש על ידי פתיחת פרוייקט ב- GitHub. את הקישור יש להעתיק ל- moodle. תשובות מילוליות יש להגיש בקובץ Word או pdf ולהעלות ל- moodle.

1. כתבי תוכנית שמדגימה את ההבדל בין class ל-struct מבחינת הסמנטיקה של ההעסקה. הראי ש-class מועבר by-reference ו-struct מועבר by value. כתבי פונקציה שמקבלת פרמטר מסוג struct ומצליחה לשנות את המשתנה שהועבר.

2. כתבי תוכנית שגורמת ל-stack overflow. תוכלי להשתמש בתוכנית הבאה (בדקי שאת מבינה כיצד התוכנית עובדת):
הגישי טבלה שמראה את הקשר בין גודל המערך `localArray` ומספר הקריאות האפשרי. הוסיפי משתנים מקומיים לפונקציה ובדקי כיצד הם משפיעים על מספר הקריאות המקסימלי (עד overflow).
הגישי טבלה עם לפחות 3 אפשרויות של משתנים מקומיים ומספר הקריאות המקסימלי.

```
public static void StackOverflowExperiment()
{
    static int RecursiveMethod(int depth, int localArraySize)
    {
        // TODO: Implement a recursive method that:
        // 1. Creates a local byte array of specified size
        byte[] localArray = new byte[localArraySize];

        // 2. Prints current recursion depth
        Console.WriteLine($"Current depth: {depth}");

        // 3. Recursively calls itself, incrementing depth
        try
        {
            return RecursiveMethod(depth + 1, localArraySize);
        }
        catch (StackOverflowException)
        {
            return depth;
        }
    }

    // Experiment with different local array sizes
    int[] arraySizes = { 100, 1000, 10000 };

    foreach (int size in arraySizes)
    {
        Console.WriteLine($"Experiment with local array size: {size} bytes");
        try
        {
            int maxDepth = RecursiveMethod(0, size);
            Console.WriteLine($"Maximum recursion depth: {maxDepth}");
        }
        catch (StackOverflowException)
        {
            Console.WriteLine("Stack overflow occurred!");
        }
    }
}
```

3. כתבי פונקציה שמחשבת גודל של אובייקטים בעזרת הפונקציה `GC.GetAllocatedBytesForCurrentThread` ניתן להשתמש בפונקציה הבאה או בפונקציה שכתבנו בכיתה **כהשראה** (זו לא התשובה!).

א. נסי הקצאה של מערכים שונים ולפי זה הסיקי מהי כמות הזיכרון הבסיסית שדרושה לכל מערך עוד לפני הזיכרון שדרוש לאיברי המערך.

ב. הקצי מערך של struct-ים בגדלים שונים – מהי השפעת גודל ה-struct על הזיכרון שמוקצה למערך.

ג. הקצי מערך של class-ים בגדלים שונים – מהי השפעת גודל כל אובייקט על הזיכרון שמוקצה למערך? הסבירי את התוצאה.

```
public static void MemoryAllocationExperiment()
{
    // Track initial allocation
    long baselineMemory = GC.GetAllocatedBytesForCurrentThread();

    // TODO: Complete the following experiments

    // 1. Allocate an array of integers
    int[] intArray = new int[10000];
    long afterIntArray = GC.GetAllocatedBytesForCurrentThread();

    // 2. Allocate an array of doubles
    double[] doubleArray = new double[10000];
    long afterDoubleArray = GC.GetAllocatedBytesForCurrentThread();

    // 3. Allocate an array of strings
    string[] stringArray = new string[10000];
    long afterStringArray = GC.GetAllocatedBytesForCurrentThread();

    // Print allocation sizes
    Console.WriteLine($"Baseline Memory: {baselineMemory} bytes");
    Console.WriteLine($"Int Array Allocation: {afterIntArray - baselineMemory} bytes");
    Console.WriteLine($"Double Array Allocation: {afterDoubleArray - afterIntArray} bytes");
    Console.WriteLine($"String Array Allocation: {afterStringArray - afterDoubleArray} bytes");
}
```

4. עייני בתוכנית הבאה :

מה התוכנית מנסה לעשות? מדוע זה לא מצליח?

כיצד ניתן לתקן את התוכנית כדי שהיא תצליח?

```
namespace ExpandArray
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] a = { 1, 2, 3 };
            for (int i = 0; i < a.Length; i++)
            {
                Console.WriteLine(a[i]);
            }
            ExpandArray(a);

            Console.WriteLine();
            for (int i = 0; i < a.Length; i++)
            {
                Console.WriteLine(a[i]);
            }
        }

        static void ExpandArray(int[] array)
        {
            int[] oldArray = array;

            array = new int[oldArray.Length * 2];
            for (int i = 0; i < oldArray.Length; i++)
            {
                array[i] = oldArray[i];
                array[i + oldArray.Length] = oldArray[i];
            }
        }
    }
}
```

בהצלחה!