РЕАЛИЗАЦИИ АЛГОРИТМОВ СОРТИРОВКИ

```
/*
            3.3 sort_0 обменом - пузырек
            3.1 sort_1 шейкерная - пузырек в обе стороны
            1.0 sort_2 выбором
            1.8 sort_3 вставкой
            3.2 sort_4 перечислением
            3.1 sort_5 гномья
            3.2 sort_6 гномья - с оптимизацией
*/

using System;
using System.Diagnostics;

class Program
{
    class _arr
    {
        public string[] name_sort =
            {"пузырьковая","шейкерная",
             "выбором","вставкой",
             "перечислением","гномья",
             "гномья с оптимизацией"};
        int[] arr;
        int ver;
        public int _ver
        {
            set { this.ver = value < 4 ? value: 0 ; }
            get { return this.ver; }
        }
        public int[] get_arr
        {
            get { return this.arr; }
        }
        public _arr()
        {   // конструктор без параметров
            arr = gen_array(10, 100);
            this.ver = 0;
        }
        public _arr(int count, int max)
        {   // конструктор с параметрами
            arr = gen_array(count, max);
            this.ver = 0;
        }
        int[] gen_array(int count, int max)
        {
            Random r = new Random((int)DateTime.Now.Ticks);
            arr = new int[count];
```

```
    for (int i = 0; i < arr.Length; i++)
        arr[i] = r.Next(max);
    return arr;
}
public void print()
{
    int i = 0;
    foreach (int a in arr)
        Console.WriteLine(++i + (char)9 + a);
    Console.WriteLine("- - -");
}
public void print(int count)
{
    if (count > arr.Length) count = arr.Length;
    for (int i = 0; i < count; i++)
        Console.Write(arr[i] + " ");
    Console.WriteLine();
}
private static void swap(ref int x, ref int y)
{
    int temp = x; x = y; y = temp;
}
public void sort()
{
    switch (ver)
    {
        case 0:
            {
                // сортировка ОБМЕНОМ
                // правую границу двигай к левой
                // пузырек всплывает вправо
                for (int j = arr.Length - 1; j > 0; j--)
                    for (int i = 0; i < j; i++)
                        if (arr[i] > arr[i + 1])
                            swap(ref arr[i],
                                 ref arr[i + 1]);
                break;
            }
        case 1:
            {
                // сортировка ШЕЙКЕРНАЯ
                // пузырек всплывает вправо
                // потом пузырек тонет влево и т.д.
                int L = 0, R = arr.Length - 1;
                while (L < R)
                {
                    for (int i = L; i < R; i++)
                        if (arr[i] > arr[i + 1])
                            swap(ref arr[i],
                                 ref arr[i + 1]);
```

```
                R--;
                for (int i = R; i > L; i--)
                    if (arr[i] < arr[i - 1])
                        swap(ref arr[i],
                             ref arr[i - 1]);
                L++;
            }
            break;
        }
    case 2:
        {
        // сортировка ВЫБОРОМ
        // левую границу двигай к правой
            for (int j = 0; j < arr.Length - 1; j++)
            {
        // найди мин и поставь на левую границу
                int mini = j;
                for (int i = j + 1; i < arr.Length; i++)
                    if (arr[i] < arr[mini])
                        mini = i;
                swap(ref arr[j], ref arr[mini]);
            }
            break;
        }
    case 3:
        {
        // сортировка ВСТАВКОЙ
        // границу отсортированного двигай вправо
            for (int j = 1; j < arr.Length; j++)
            {   // найди место куда вставить
                for (int i = j; i > 0; i--)
                    if (arr[i - 1] > arr[i])
                        swap(ref arr[i - 1],
                             ref arr[i]);
                    else break;
            }
            break;
        }
    case 4:
        {
// сортировка ПЕРЕЧИСЛЕНИЕМ - найти рейтинг каждого
// сравнить попарно все и
// найти сколько меньше каждого
            int[] reyting = new int[arr.Length];
// массив рейтингов
            int[] rey = new int[arr.Length];
// массив временный
            for (int i=0; i<reyting.Length; i++)
                reyting[i] = 0;
            for (int j = 0; j < arr.Length-1; j++)
```

```
            for (int i=j+1; i<arr.Length; i++)
                if (arr[j] > arr[i])
                    reyting[j]++;
                else
                    reyting[i]++;
        for (int i=0; i < arr.Length; i++)
            rey[reyting[i]] = arr[i];
        arr = rey;
        break;
    }
case 5:
    {
    // сортировка ГНОМЬЯ - начни слева
        // если менять не надо, двигай вправо
        int i = 1;
        while (i<arr.Length)
        {
            if (arr[i-1] < arr[i])
                i++;
            else
            {
                swap(ref arr[i-1],
                     ref arr[i]);
                if (i>0) i--;
            }
        }
        break;
    }
case 6:
    {
        // сортировка ГНОМЬЯ - с оптимизацией
        // запоминаем место, где сменили курс
        int i = 1, flag = 0;
        while (i < arr.Length)
        {
            if (arr[i-1] < arr[i])
            {
                i = flag > i ? flag+1 : i+1;
            }
            else
            {
                swap(ref arr[i-1],
                     ref arr[i]);
                if (flag < i) flag = i;
                if (i > 1) i--;
            }
        }
        break;
    }
default:
```

```
                {
                    Console.WriteLine("Нет метода...");
                    break;
                }
            }
        }
    }

    static Stopwatch sw = new Stopwatch();
    static void print_time()
    {
        TimeSpan ts = sw.Elapsed;
        string elapsedTime =
               String.Format("{0:00}:{1:00}.{2:000}",
               ts.Minutes, ts.Seconds, ts.Milliseconds);
        Console.WriteLine("  Duration of work - " +
                          elapsedTime);
    }

    static void Main(string[] args)
    {
        int len = 20000;     // кол-во элементов массива
        int max = 5;        // размерность элементов
        int met = 7;        // сколько методов сортировки
        for (int i=0; i<met; i++)
        {
            _arr p = new _arr(len, max);
            p._ver = i; // устанавливаем метод сортировки
            Console.WriteLine(p.name_sort[i]); // имя метода
            p.print(10);
            sw.Restart();
            p.sort();
            sw.Stop();
            print_time();
            p.print(10);
        }

        Console.ReadKey();
        /*
            методы с внутренней сортировкой, кроме 3
            3.3 sort_0 обменом - пузырек
            3.1 sort_1 шейкерная - пузырек в обе стороны
            1.0 sort_2 выбором
            1.8 sort_3 вставкой
            3.2 sort_4 перечислением
            3.1 sort_5 гномья
            3.2 sort_6 гномья - с оптимизацией */
    }
}
```