

**ОТЧЕТ О ВЫПОЛНЕНИИ**

**практической работы № 8**

Выполнил: ст.гр.

2ИСП9-23

Быховцева Дарья

Специальность: 09.02.07

Информационные системы и программирование

Проверил: Боклач Б.И.

Москва 2022

Методы сортировки. Вариант 7.

Задание 1.

Сортировка выбором.

Выбираем в массиве элемент с минимальным значением на интервале от 1-го до n-го элемента и меняем его местами с первым элементом.

static void Main(string[] args)

{

Console.WriteLine("Ведите количество чисел для сортировки:");

int N = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите числа для сортировки:");

int[] mas = new int[N];

for (int i = 0; i < mas.Length; i++)

{

mas[i] = Convert.ToInt32(Console.ReadLine());

}

Sortirovka(mas);

Console.WriteLine("Отсортированный массив:");

for (int i = 0; i < mas.Length; i++)

{

Console.WriteLine(mas[i]);

}

Console.ReadLine();

}

static int[] Sortirovka(int[] mas) //создание функции сортировки

{

for (int i = 0; i < mas.Length - 1; i++)

{

//поиск минимального числа

int min = i;

for (int j = i + 1; j < mas.Length; j++)

{

if (mas[j] < mas[min])

{

min = j;

}

}

//обмен элементов

int m = mas[min];

mas[min] = mas[i];

mas[i] = m;

}

return mas;

}

Сортировка вставкой.

Массив делится на две части – отсортированная и не отсортированная. Элементы из не отсортированной части вставляются в отсортированную часть на определенные позиции в массиве(так, чтобы не нарушить упорядоченность элементов). В начале работы алгоритма в качестве отсортированной части массива принимают только один первый элемент, а в качестве не отсортированной части - все остальные элементы.

static void Main(string[] args)

{

int[] a = { 7, 0, -4, 3, 1, -2, 5 };

for (int i = 1; i < a.Length; i++)

{

int k = a[i];

int j = i - 1;

while (j >= 0 && a[j] > k)

{

a[j + 1] = a[j];

j--;

}

a[j + 1] = k;

}

Console.WriteLine("Сортировка вставками");

for (int i = 0; i < a.Length; i++)

{

Console.WriteLine(a[i]);

}

Console.ReadKey();

}

Сортировка методом пузырька.

Попарное сравнение элементов, начиная с нулевого в массиве.

static void Main(string[] args)

{

Console.WriteLine("Введите количество чисел для сортировки:");

int N = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите числа для сортировки:");

int[] mas = new int[N];

for (int i = 0; i < mas.Length; i++)

{

mas[i] = Convert.ToInt32(Console.ReadLine());

}

SortirovkaPoop(mas);

Console.WriteLine("После сортировки:");

for (int i = 0; i < mas.Length; i++)

{

Console.WriteLine(mas[i]);

}

Console.ReadLine();

}

static int[] SortirovkaPoop(int[] mas)

{

int m;

for (int i = 0; i < mas.Length; i++)

{

for (int j = i + 1; j < mas.Length; j++)

{

if (mas[i] > mas[j])

{

m = mas[i];

mas[i] = mas[j];

mas[j] = m;

}

}

}

return mas;

}

Задание 2.

Создать массив целочисленного типа А(n) и выполнить поиск заданного элемента s методом двоичного поиска. На экран вывести найденный элемент, либо запись, что элемент не найден.

using System;

class GFG – метод наследования(гикфоркигс). Одному классу разрешено наследовать функции другого класса.

{

public static void Main(String[] args)

{

int[] arr = new int[7] { 1, 5, 7, 4, 6, 2, 3 };

Array.Sort(arr);

Console.Write("Элементы массива: ");

display(arr);

object s = 8; - Поддерживает все классы в иерархии классов .NET и предоставляет низкоуровневые службы для производных классов. Является исходным базовым классом для всех классов .NET и корнем иерархии типов.

result(arr, s);

object s1 = 4;

result(arr, s1);

}

static void result(int[] arr2, object k)

{

int res = Array.BinarySearch(arr2, k);

if (res < 0)

{

Console.WriteLine("\nЭлемент "

+ "({0}) не найден.",

k);

}

else

{

Console.WriteLine("Элемент "

+ "({0}) найден {1}.",

k, res);

}

}

static void display(int[] arr1)

{

foreach (int i in arr1)

Console.Write(i + " ");

}

}

