

Список используемых источников:

1. ГОСТ 19.003-80. Схемы алгоритмов и программ. – Москва : Изд-во стандартов, 1982. – 27 с.
2. Жибинова И.А. Оформление учебных, научных, проектных работ студентов: методические указания / И. А. Жибинова, С.Р. Зельцер, А.Н. Жибинов, О.В. Михайлова – Новокузнецк: НФИ КемГУ, 2012. – 91 с.
3. Буйначев С.К. Основы программирования на языке Python : учебное пособие / С.К. Буйначев, Н.Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91, [1] с.+
4. Разумников С.В. Теория алгоритмов: методические указания к выполнению курсовой работы для студентов очной формы обучения, обучающихся по направлению 09.03.03 «Прикладная информатика» / С.В. Разумников – Юрга: Изд-во Юргинского технологического института (филиала) Томского политехнического университета, 2022. – 20 с.

СОРТИРОВКА ОДНОМЕРНОГО МАССИВА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C#

*Д.С. Родионова², ученица 7 класса,
Научный руководитель: Разумников С.В.^{1а}, к.т.н., доц.,
¹Юргинский технологический институт (филиал)
Национального Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
²МБОУ Средняя общеобразовательная школа № 14 имени К.С. Федоровского
652055, Кемеровская обл., г. Юрга, ул. Волгоградская, 3
E-mail: ^arazumnikov@tpu.ru*

Аннотация. В статье рассматривается способ сортировки одномерного массива на языке программирования C#. Приведен пример реализации с входными и выходными данными.

Ключевые слова: программирование, C#, одномерный массив, сортировка.

Annotation. The article discusses a way to sort a one-dimensional array in the C# programming language. An example implementation with input and output data is given.

Keywords: programming, C#, one-dimensional array, sorting.

Введение. В программировании очень часто появляется потребность в упорядочивании коллекции данных. Будь то список учеников по алфавиту, сводные таблицы результативности, или еще какие-нибудь статистические данные. Да все что угодно можно упорядочить. Главное выбрать критерий, по которому необходимо проводить сортировку. Наиболее распространена сортировка по возрастанию и убыванию значений, или сортировка по алфавиту.

Чтобы провести сортировку по определенному критерию, нужно понять алгоритм упорядочивания. Их много, одни быстрые и сложные в написании, другие медленнее, но простые в реализации. В этой статье мы разберем самый популярный метод сортировки – сортировка методом пузырька.

Сортировка методом пузырька, самая простая в реализации, но одна из самых долгих в исполнении, при самом худшем случае, при количестве элементов n , время выполнения сортировки будет n^2 . Поэтому используется только в учебных целях и на практике почти никогда не применяется.

Само название метода говорит о его способе реализации. Как самые большие пузыри, опережая средние и маленькие пузырьки в воде выходят на поверхность, так и в массиве – самые крупные значения опережают более меньшие числа.

Для того чтобы отсортировать по возрастанию, мы будем поочередно сравнивать пару элементов, и если левый элемент больше правого, то менять их местами. Рассмотрим на примере сортировки массива `array` по возрастанию:

```
int[] array = {24, 19, 18, 30, 17, 26};
```

Распишем трассировочную таблицу при каждой итерации сравнения:

Сравниваем элемент массива с индексом 0 поочередно со следующими элементами

Шаг 1.	24 19 18 30 17 26	сравнение 24 и 19
	19 24 18 30 17 26	обмен 24 19
Шаг 2.	19 24 18 30 17 26	сравнение 19 и 18
	18 24 19 30 17 26	обмен 19 18

XIV Всероссийская научно-практическая конференция
для студентов и учащейся молодежи
«Прогрессивные технологии и экономика в машиностроении»

Шаг 3. 18 24 19 30 17 26 сравнение 18 и 30
18 24 19 30 17 26 не меняем
Шаг 4. 18 24 19 30 17 26 сравнение 18 и 17
17 24 19 30 18 26 обмен 18 17
Шаг 5. 17 24 19 30 18 26 сравнение 17 и 26
17 24 19 30 18 26 не меняем
Сравниваем элемент массива с индексом 1 поочередно со следующими элементами
Шаг 6. 17 24 19 30 18 26 сравнение 24 и 19
17 19 24 30 18 26 обмен 24 19
Шаг 7. 17 19 24 30 18 26 сравнение 19 и 30
17 19 24 30 18 26 не меняем
Шаг 8. 17 19 24 30 18 26 сравнение 19 и 18
17 18 24 30 19 26 обмен 19 18
Шаг 9. 17 18 24 30 19 26 сравнение 18 и 26
17 18 24 30 19 26 не меняем
Сравниваем элемент массива с индексом 2 поочередно со следующими элементами
Шаг 10. 17 18 24 30 19 26 сравнение 24 и 30
17 18 24 30 19 26 не меняем
Шаг 11. 17 18 24 30 19 26 сравнение 24 и 19
17 18 19 30 24 26 обмен 24 19
Шаг 12. 17 18 19 30 24 26 сравнение 19 и 26
17 18 19 30 24 26 не меняем
Сравниваем элемент массива с индексом 3 поочередно со следующими элементами
Шаг 13. 17 18 19 30 24 26 сравнение 30 и 24
17 18 19 24 30 26 обмен 30 24
Шаг 14. 17 18 19 24 30 26 сравнение 24 и 26
17 18 19 24 30 26 не меняем
Сравниваем элемент массива с индексом 4 поочередно со следующими элементами
Шаг 15. 17 18 19 24 30 26 сравнение 30 и 26
17 18 19 24 26 30 обмен 30 26

```
//Листинг решения пузырьковой сортировки
using System;
namespace Example {
class Program {
static void Main() {
int[] array = { 24, 19, 18, 30, 17, 26 };
for (int i = 0; i < array.Length - 1; i++)
for (int j = i + 1; j < array.Length; j++)
if (array[i] > array[j]) {
int temp = array[i];
array[i] = array[j];
array[j] = temp;
}

Console.WriteLine(string.Join(" ", array));
Console.ReadKey();
}
}
```

Пример

С консоли дан неупорядоченный массив данных роста учеников в сантиметрах. А также рост нового ученика. Напишите программу, которая упорядочит значения по убыванию, и укажет, на каком месте расположится новый ученик в шеренге по росту на уроке физкультуры. Не использовать метод `Array.Sort!`

Пример использования:

Пример использования:

Выходные данные 1: Введите рост учеников, перечисляйте значения через пробел.

Входные данные 1: 164 150 172 182 169 190 184 164 162 176 181 179 182

Выходные данные 1: Упорядоченный по росту массив:

XIV Всероссийская научно-практическая конференция
для студентов и учащейся молодежи
«Прогрессивные технологии и экономика в машиностроении»

```
Выходные данные 1: 190 184 182 182 181 179 176 172 169 164 164 162 150
Выходные данные 1: Введите рост нового ученика
Входные данные 1: 178
Выходные данные 1: Позиция в строю на №7
}
}
}
```

Пример

Решение:

Задача очень проста в решении, если не отлавливать критические ситуации представленные в примерах использования.

Принимаем строку от пользователя, проверим его на пустое значение, если строка равна `string.Empty`, то завершим приложение с выводом сообщения, что был пустой ввод. Иначе, в безопасной конструкции `try/catch` попробуем преобразовать строку в числовой массив, если произойдет ошибка – то сообщим о не корректной конвертации и выйдем из приложения. Иначе, отсортируем массив по убыванию, где в начале массива будут находиться максимальные значения и убывать по мере увеличения индекса элемента. Сортировать будет методом пузырька.

Далее, приглашаем ввести числовое значение роста ученика, проверим его на правильный ввод, и если будет ошибка преобразования – выйдем из программы с сообщением о не корректном вводе числа. Иначе, организуем цикл `for` для перебора всего массива упорядоченного массива, где в теле, создадим условие, в котором спросим, если рост больше текущего элемента массива, то закончим цикл и сделаем вывод в консоль номер позиции ученика в строю – переменная цикла плюс единица.

Но в этой задаче может быть ситуация, когда рост ученика будет меньше роста всех учеников, поэтому нам этот момент необходимо тоже учитывать.

```
//Листинг решения задачи array_37
using System;
namespace Example {
class Program {
static void Main() {
Console.WriteLine("Введите рост учеников, перечисляйте значения через пробел.");
string input = Console.ReadLine().Trim();
if (input != "") {
int[] array;
try {
array = Array.ConvertAll(input.Split(' '), int.Parse);
} catch {
Console.WriteLine("Массив введен не корректно!");
Console.ReadKey();
return;
}
for (int i = 0; i < array.Length - 1; i++)
for (int j = i + 1; j < array.Length; j++)
if (array[i] < array[j]) {
int temp = array[i];
array[i] = array[j];
array[j] = temp;
}
Console.WriteLine("Упорядоченный по росту массив:\n" + string.Join(" ", array));
Console.WriteLine("Введите рост нового ученика");
if (!int.TryParse(Console.ReadLine().Trim(), out int height)) {
Console.WriteLine("Не корректный рост нового ученика!");
Console.ReadKey();
return;
}
for (int i = 0; i < array.Length; i++) {
```

```
if (height > array[i]) {  
    Console.WriteLine("Позиция в строку на №" + (i + 1));  
    Console.ReadKey();  
    return;  
}  
}  
Console.WriteLine("Позиция в строку на №" + (array.Length + 1));  
} else Console.WriteLine("Ошибка ввода. Пустая строка");  
Console.ReadKey();  
}  
}
```

Список используемых источников:

1. Разумников С.В. Алгоритм и программное обеспечение для построения агрегированных рейтингов / С.В. Разумников // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2021. – т. 24 – № 2. – С. 39–44.
2. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования / С.В. Разумников // Вестник ВГУ, серия: Системный анализ и информационные технологии. – 2021. – № 2. – С. 138–152.

СИСТЕМНЫЙ АНАЛИЗ ПРОЦЕССА РАЗРАБОТКИ КОРПОРАТИВНОГО САЙТА В WEB-СТУДИИ

А.А. Лузинсан, студент гр. 430-2

научный руководитель: Захарова А.А., проф., д.т.н, доц.

*Томский государственный университет систем управления и радиоэлектроники,
634045, Томская обл., г. Томск, ул. Ленина, 40*

E-mail: uzinsan@mail.ru

Аннотация: в связи с текущей ситуацией в мире встаёт вопрос создания инструмента для рационального принятия решений в процессе разработки информационных продуктов, в том числе корпоративного сайта. В текущей статье представлены результаты исследования альтернатив решения проблемы, касаемой превышения бюджета на разработку корпоративного сайта.

Ключевые слова: системный анализ, корпоративный сайт, web-студия, превышение бюджета, метод группового парного сравнения.

Abstract: In light of the current situation in the world, the question arises of creating a tool for rational decision-making in the development of information products, including a corporate website. This article presents the results of research on alternatives to addressing the problem of exceeding the budget for developing a corporate website.

Keywords: Systematic analysis, corporate website, web studio, budget overrun, paired comparison method.

В данной работе рассматривается проблемная ситуация, заключающаяся в превышении бюджета на разработку корпоративного сайта web-студией. Актуальность исследования указанной проблемы затрагивает не только предпринимателя, желающего заказать интернет-ресурс, но и web-студию, которая должна каждый раз трезво оценивать свои возможности по выполнению предлагаемого проекта.

Среди факторов, собранных в модели чёрного ящика (рис.1), которые так или иначе влияют на эффективность разработки, были выделены: интернет-аудитория; научно-исследовательские институты, предоставляющие новейшие технологии, которые определяют актуальность разрабатываемых продуктов; целевая аудитория как подмножество интернет-пользователей, которые будут взаимодействовать с конечным продуктом; заказчик – непосредственный источник дохода, целей, задач и материалов для разработки; конкуренты представлены прочими web-студиями, которые предоставляют альтернативные решения; поставщики технологических и производственных ресурсов, обеспечивающие оборудованием и инструментами разработки; рынок труда и Outsourse, использующийся в ряде специфических случаев, и другие акторы.