

# *Сортировка простым обменом. (методом «пузырька»)*

Рассмотрим идею метода на примере.

Отсортируем по возрастанию массив  
из 5 элементов:

5 4 8 2 9

# Первый просмотр

рассматривается весь массив:

**i=1**            **5**    **4**    **8**    **2**    **9**

> меняем

**i=2**            **4**    **5**    **8**    **2**    **9**

< не меняем

**i=3**            **4**    **5**    **8**    **2**    **9**

> меняем

**i=4**            **4**    **5**    **2**    **8**    **9**

< не меняем

**9** находится на своем месте.

## Второй просмотр

рассматривается часть массива с первого до предпоследнего элемента:

$i=1$       4      5      2      8      9

< не меняем

$i=2$       4      5      2      8      9

> меняем

$i=3$       4      2      5      8      9

< не меняем

**8** — на своем месте.

## Третий просмотр

рассматривается часть массива, содержащая три первых элемента:

$i=1$             **4**    **2**    5    8    9

> меняем

$i=2$             2    **4**    **5**    8    9

< не меняем

**5** — на своем месте.

---

## Четвертый просмотр

рассматривается последняя пара элементов:

$i=1$             **2**    **4**    5    8    9

< не меняем    **4** - на своем месте.

Наименьший элемент — **2** оказывается на первом месте.

Количество просмотров элементов массива равно  $N-1$   
Этот метод также называют методом «пузырька».  
Название это происходит от образной интерпретации,  
при которой в процессе выполнения сортировки более  
«легкие» элементы (элементы с заданным свойством)  
мало-помалу всплывают на «поверхность».

***Var***

***k, i, w: Integer;*** {*k* - номер просмотра, изменяется от 1 до *N-1*;  
*i* - номер первого элемента рассматриваемой пары;  
*w* - рабочая переменная для перестановки местами элементов массива.}

***Begin***

***For k:=1 To N-1 Do*** {Цикл по номеру просмотра. }

***For i:=1 To N-k Do***

***If A[i]>A[i+1] Then*** {'Перестановка элементов.}

***Begin***

***w:=A[i];***

***A[i] :=A[i+1];***

***A[i+1] :=w;***

***End;***

***End;***

При сортировке методом «пузырька» выполняется *N-1* просмотров, на каждом *i*-просмотре производится *N-i* сравнений.





