



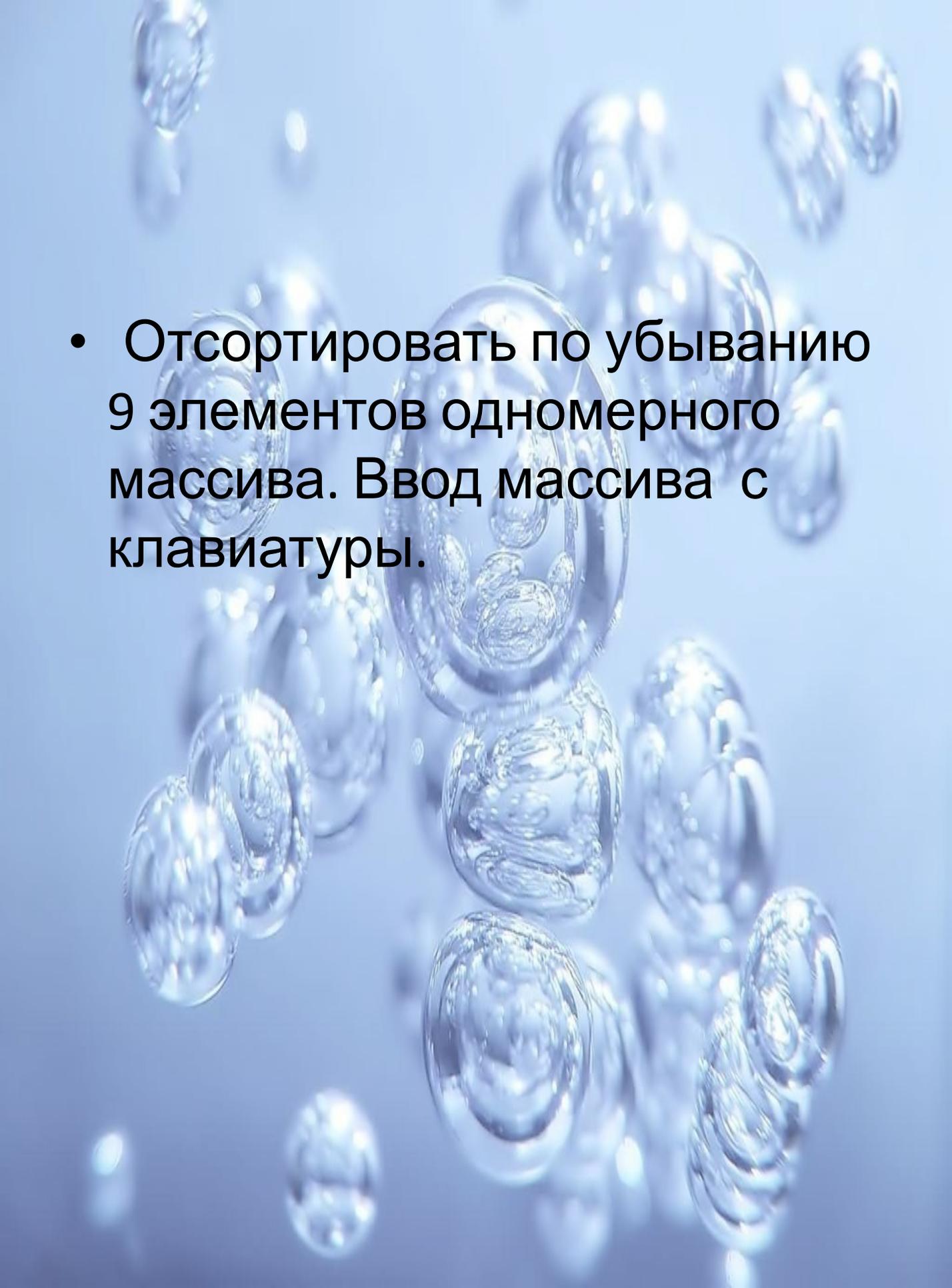
Сортировка массива

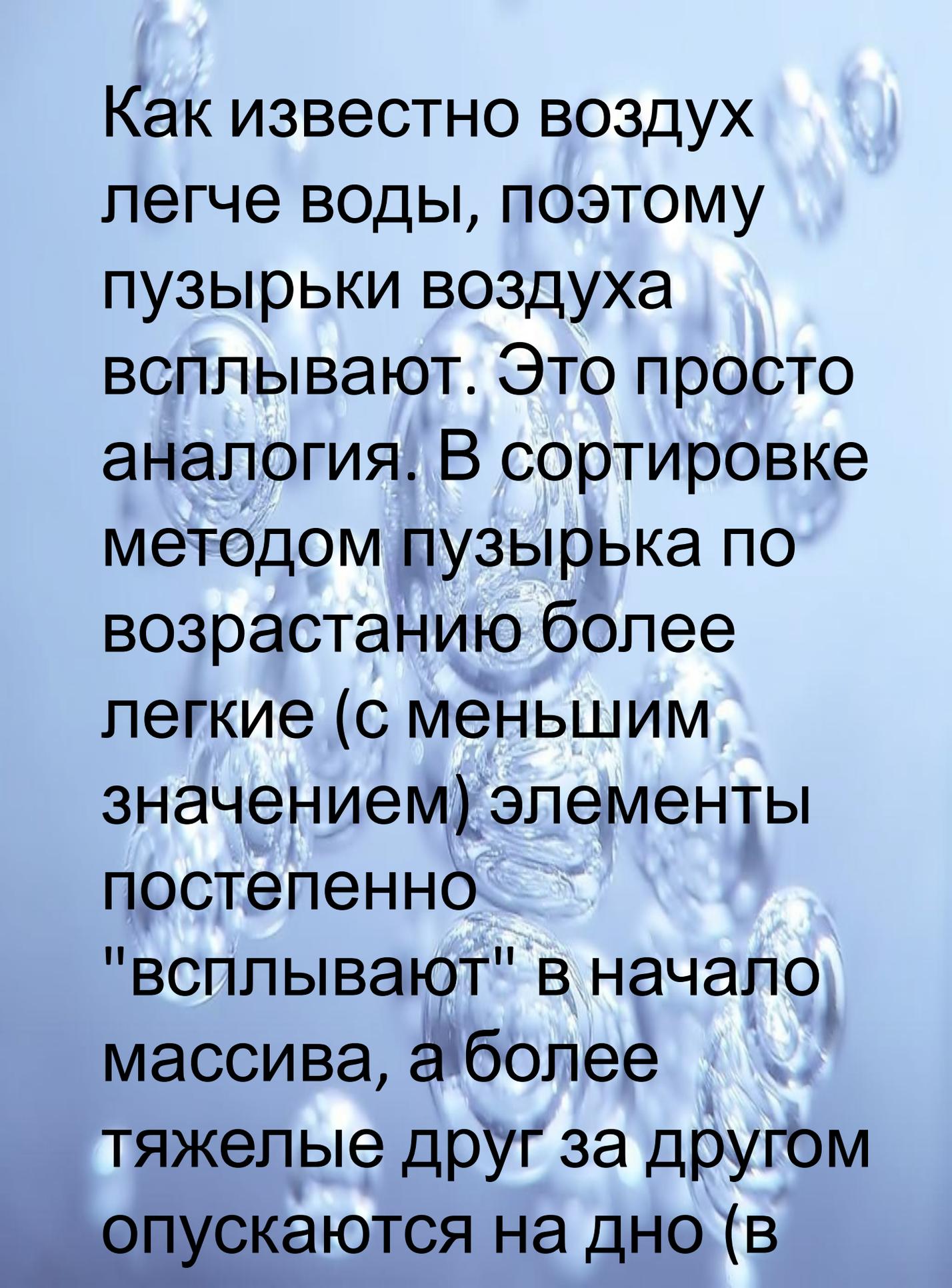
Метод пузырька

- При работе с массивами данных не редко возникает задача их **сортировки по возрастанию или убыванию, т.е. упорядочивания**. Это значит, что элементы того же нужно расположить строго по порядку. Например, в случае сортировки по возрастанию предшествующий элемент должен быть меньше



Достаточно простой
для понимания
является сортировка
методом пузырька,
который также
называют **методом**
простого обмена.

- 
- Отсортировать по убыванию 9 элементов одномерного массива. Ввод массива с клавиатуры.



Как известно воздух легче воды, поэтому пузырьки воздуха всплывают. Это просто аналогия. В сортировке методом пузырька по возрастанию более легкие (с меньшим значением) элементы постепенно "всплывают" в начало массива, а более тяжелые друг за другом опускаются на дно (в

- **const** m = 10;
- **var** arr: **array**[1..m] **of integer**;
- i, j, k: **integer**;
- **begin**
- randomize;
- write ('Исходный массив: ');
- **for** i := 1 **to** m **do**
- **begin** arr[i] := random(256);
- write (arr[i]:4);
- **end**;
- writeln;
- writeln;
- **for** i := 1 **to** m-1 **do**
- **for** j := 1 **to** m-i **do if** arr[j] > arr[j+1] **then**
- **begin** k := arr[j]; arr[j] := arr[j+1]; arr[j+1] := k
- **end**;
- write ('Отсортированный массив: ');
- **for** i := 1 **to** m **do** write (arr[i]:4);
- writeln;
- Readln
- **end.**

Повторение алгоритма сортировки

- [Алгоритм](#)

Задача

Отсортировать по возрастанию 20 элементов одномерного массива. Ввод массива осуществить любым способом.

Алгоритм

- **Программа.** Объявить и сформировать массив.
- Отобразить исходный массив.
- Для каждого J от 1 до $N-1$ повторять:
- Для каждого I от 1 до $N-1$ повторять:
- Если $M(I) > M(I+1)$ то:
- Присвоить V значение $M(I)$
- Присвоить $M(I)$ значение $M(I+1)$
- Присвоить $M(I+1)$ значение V
- Конец Если
- Конец Цикла по I
- Конец Цикла по J
- Отобразить отсортированный массив.
- Конец Алгоритма