

ОТЧЕТ
по практической работе №1
по курсу «Объектно-ориентированное программирование»
Тема: «Разработка класса "Массив" для работы с вещественными числами»

Выполнил:
студенты группы 4375
Газизов А.А.

Санкт-Петербург
2025

Содержание

1	Задание на практическую работу	2
2	Спецификации классов	3
2.1	Спецификация класса Array	3
3	Диаграмма классов	4
4	Описание контрольного примера	4
4.1	Входные данные:	4
4.2	Ожидаемые результаты:	4
5	Скриншоты работы программы	5
6	Выводы по выполнению работы	8

1 Задание на практическую работу

Разработать класс "Массив способный хранить объекты абстрактного типа **number**.
Класс должен уметь:

- Создавать пустой массив (без элементов)
- Создавать массив с N элементами и заполнять их заданными значениями
- Обеспечивать корректное уничтожение объектов класса
- Изменять размерность массива
- Вводить элементы массива из стандартного потока ввода
- Выводить элементы массива в стандартный поток вывода
- Вычислять среднее и среднеквадратичное отклонение (СКО)
- Сортировать элементы массива по возрастанию и убыванию

Создать консольное приложение согласно представленной диаграмме классов, предназначенное для работы с массивом вещественных чисел.

2 Спецификации классов

2.1 Спецификация класса Array

Таблица 1: Первичный протокол класса Array

Атрибуты			
Идентификатор	Тип	Область видимо- сти	Семантическое описание
data	number*	private	Указатель на динамический массив элементов типа number
capacity	int	private	Реальная емкость выделенной памяти
size	int	private	Фактическое количество элементов в массиве
Методы			
Идентификатор	Область видимости		Семантическое описание
Array()	public		Конструктор по умолчанию, создает пустой массив
Array(int n, number value = 0)	public		Параметризированный конструктор, создает массив из n элементов
Array()	public		Деструктор, освобождает выделенную память
getSize() const	public		Возвращает текущий размер массива
operator[] (int index)	public		Оператор доступа к элементу по индексу (для изменения)
operator[] (int index) const	public		Оператор доступа к элементу по индексу (только для чтения)
changeSize(int newSize, number value = 0)	public		Изменяет размер массива
inputFromStream(std::istream& in)	public		Ввод данных из потока
outputToStream(std::ostream& out) const	public		Вывод данных в поток
calculateAverage() const	public		Вычисление среднего значения
calculateStandardDeviation() const	public		Вычисление среднеквадратичного отклонения
sort(bool ascending = true)	public		Сортировка массива (по возрастанию или убыванию)
resize(int newCapacity)	private		Внутренний метод для изменения емкости массива

3 Диаграмма классов

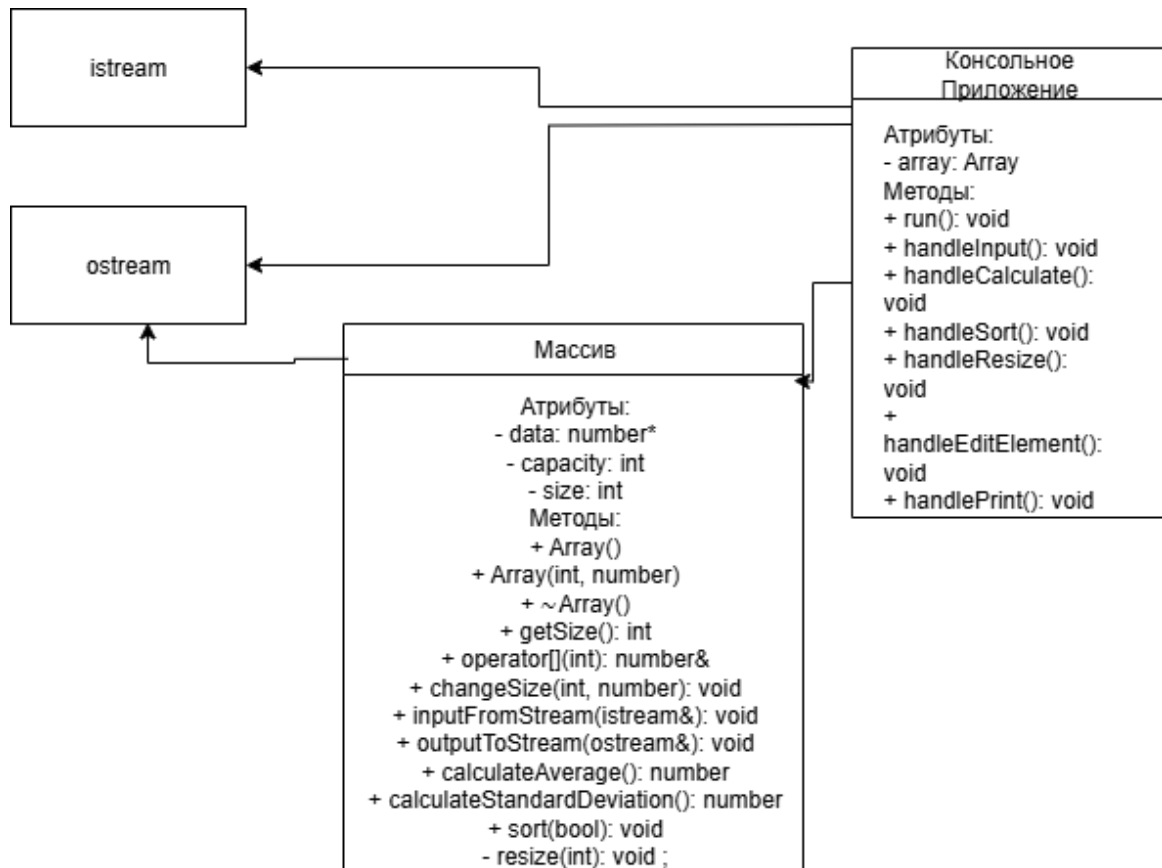


Рис. 1: Диаграмма классов для практической работы №1

4 Описание контрольного примера

Для тестирования программы был разработан контрольный пример с следующими параметрами:

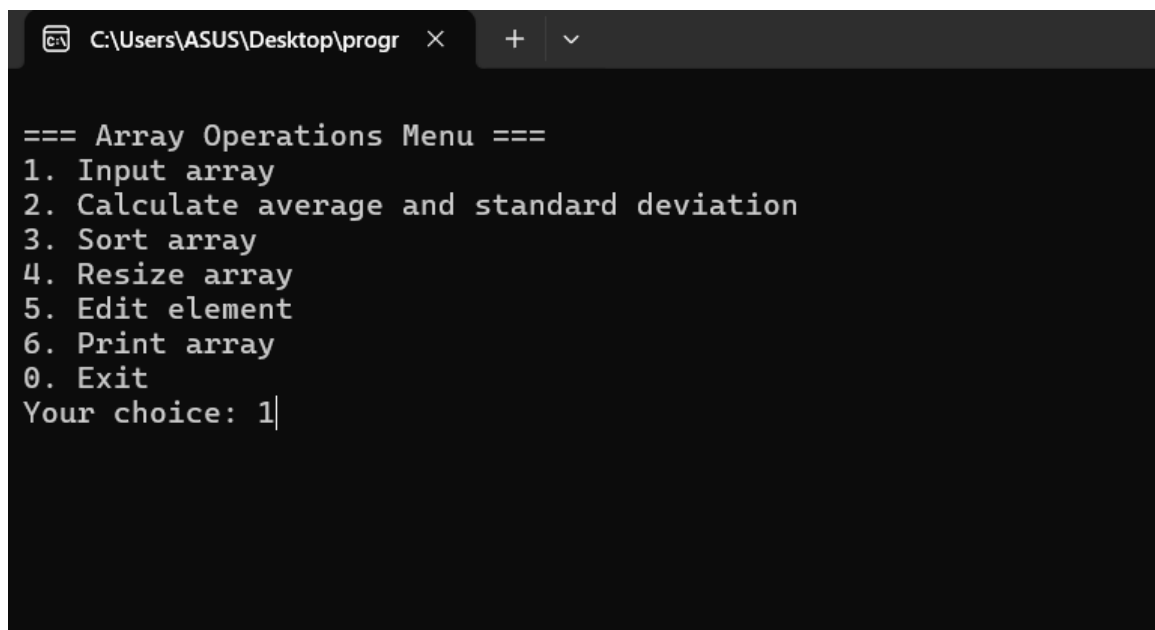
4.1 Входные данные:

- Массив: [3, 1, 4, 1, 5, 9, 2, 6]

4.2 Ожидаемые результаты:

- Среднее значение: $\frac{3+1+4+1+5+9+2+6}{8} = \frac{31}{8} = 3.875$
- СКО: $\sqrt{\frac{(3-3.875)^2+(1-3.875)^2+(4-3.875)^2+(1-3.875)^2+(5-3.875)^2+(9-3.875)^2+(2-3.875)^2+(6-3.875)^2}{8}} \approx 2.571$
- Сортировка по возрастанию: [1, 1, 2, 3, 4, 5, 6, 9]
- Сортировка по убыванию: [9, 6, 5, 4, 3, 2, 1, 1]

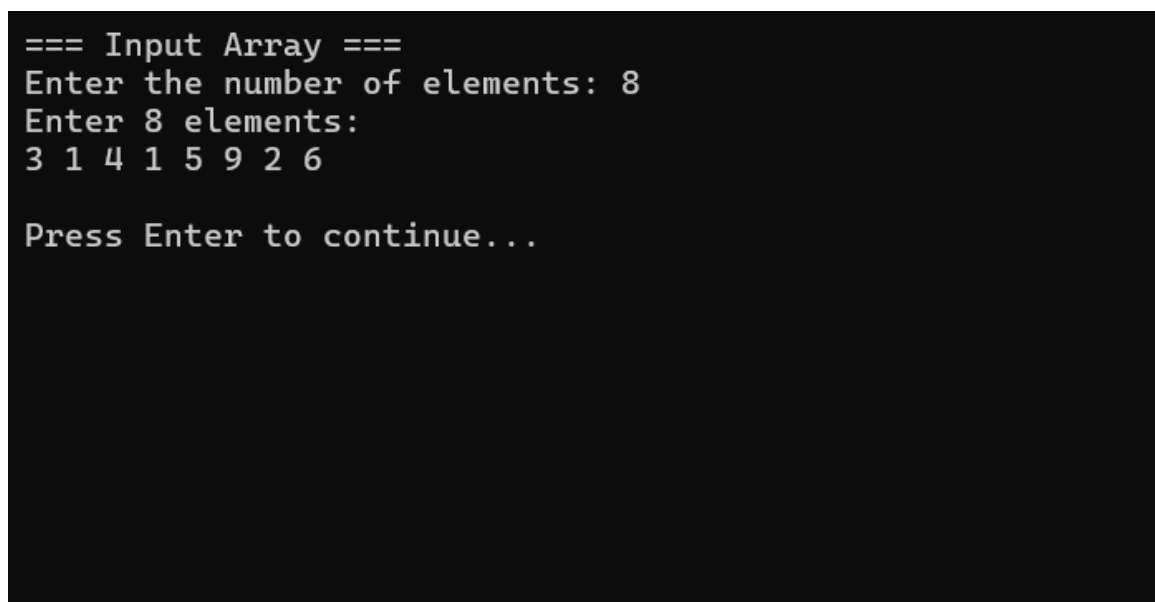
5 Скриншоты работы программы



```
C:\Users\ASUS\Desktop\progr  X  +  v

=== Array Operations Menu ===
1. Input array
2. Calculate average and standard deviation
3. Sort array
4. Resize array
5. Edit element
6. Print array
0. Exit
Your choice: 1|
```

Рис. 2: Главное меню программы



```
=== Input Array ===
Enter the number of elements: 8
Enter 8 elements:
3 1 4 1 5 9 2 6

Press Enter to continue...
```

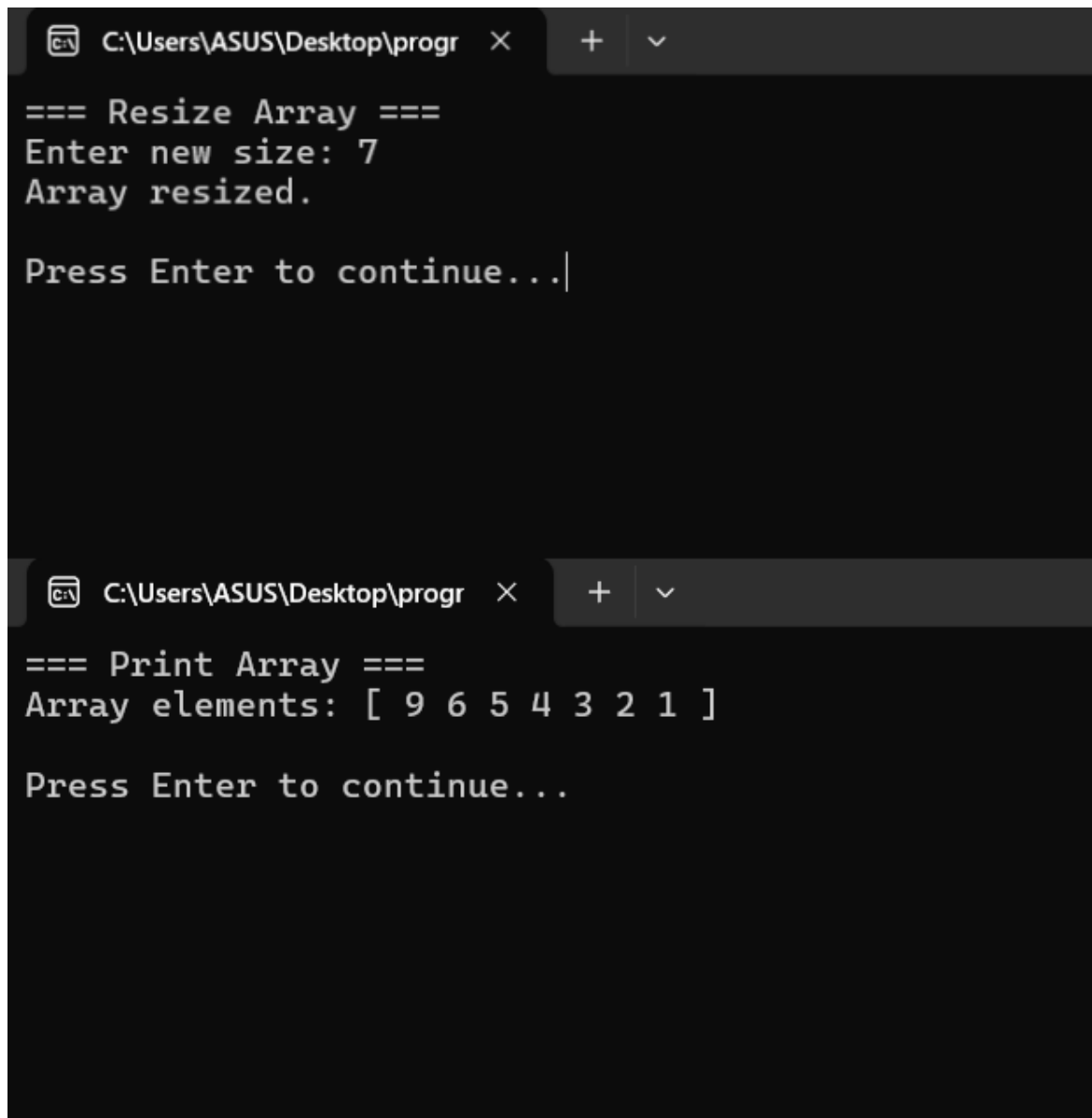
Рис. 3: Ввод элементов массива

```
C:\Users\ASUS\Desktop\progr  ×  +  ∨  
=== Calculate Average and Standard Deviation ===  
Average: 3.875  
Standard Deviation: 2.57087  
  
Press Enter to continue...
```

Рис. 4: Результаты вычислений среднего и СКО

```
C:\Users\ASUS\Desktop\progr  ×  +  ∨  
=== Sort Array ===  
Choose order (1 - Ascending, 2 - Descending): 2  
Array sorted.  
Array elements: [ 9 6 5 4 3 2 1 1 ]  
  
Press Enter to continue...|  
  
C:\Users\ASUS\Desktop\progr  ×  +  ∨  
=== Sort Array ===  
Choose order (1 - Ascending, 2 - Descending): 1  
Array sorted.  
Array elements: [ 1 1 2 3 4 5 6 9 ]  
  
Press Enter to continue...|
```

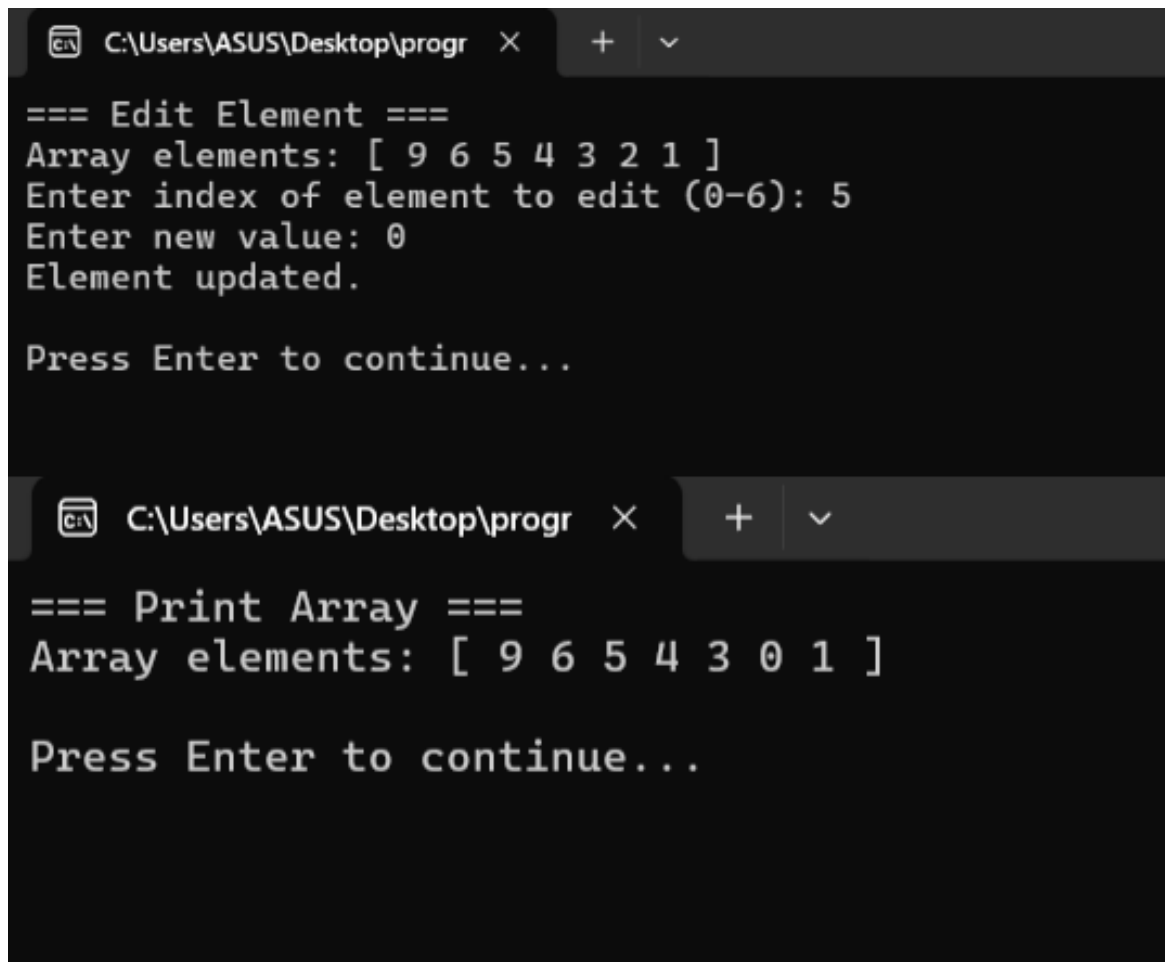
Рис. 5: Результаты сортировки массива



The image displays two sequential screenshots of a C# console application. The top screenshot shows the 'Resize Array' section where the user enters '7' as the new size, and the array is successfully resized. The bottom screenshot shows the 'Print Array' section where the array elements [9, 6, 5, 4, 3, 2, 1] are printed, confirming the new size of 7.

```
C:\Users\ASUS\Desktop\progr X + v  
=== Resize Array ===  
Enter new size: 7  
Array resized.  
  
Press Enter to continue...|  
  
C:\Users\ASUS\Desktop\progr X + v  
=== Print Array ===  
Array elements: [ 9 6 5 4 3 2 1 ]  
  
Press Enter to continue...
```

Рис. 6: Результаты изменения размера массива



The image contains two screenshots of a C# console application. The top screenshot shows the 'Edit Element' function. It displays an array [9, 6, 5, 4, 3, 2, 1] and prompts the user to enter an index (0-6) and a new value. The user enters index 5 and value 0, resulting in the array [9, 6, 5, 4, 3, 0, 1]. The bottom screenshot shows the 'Print Array' function, which displays the updated array [9, 6, 5, 4, 3, 0, 1]. Both screenshots show the file path C:\Users\ASUS\Desktop\progr and window controls.

```
=== Edit Element ===  
Array elements: [ 9 6 5 4 3 2 1 ]  
Enter index of element to edit (0-6): 5  
Enter new value: 0  
Element updated.  
  
Press Enter to continue...  
  
=== Print Array ===  
Array elements: [ 9 6 5 4 3 0 1 ]  
  
Press Enter to continue...
```

Рис. 7: Результаты изменения элемента по индексу

6 Выводы по выполнению работы

В ходе выполнения практической работы №1 были достигнуты следующие результаты:

1. Разработан класс **Array**, инкапсулирующий логику работы с динамическим массивом элементов типа **number**.
2. Реализованы все требуемые функциональные возможности:
 - Создание массива различными способами
 - Изменение размера массива
 - Ввод/вывод данных с проверкой корректности ввода
 - Вычисление статистических характеристик
 - Сортировка элементов
 - Изменение элемента массива по индексу
3. Обеспечена абстракция от конкретного типа данных через использование псевдонима **number**, что позволяет легко изменить тип элементов массива в будущем.

4. Создано консольное приложение с интуитивно понятным меню для взаимодействия с пользователем. Реализована очистка экрана и проверка ввода для всех пунктов меню.
5. Протестирована работа программы на контрольном примере, результаты совпали с ожидаемыми.
6. Программа реализована в соответствии с принципами объектно-ориентированного программирования:
 - Инкапсуляция: данные и методы работы с ними объединены в классы
 - Абстракция: использование псевдонима `number` для абстрагирования от конкретного типа
 - Перегрузка операторов: для удобного доступа к элементам массива

Основные улучшения по сравнению с первоначальной версией:

- Добавлена проверка ввода для всех операций
- Реализована очистка экрана перед выводом меню и результатов
- Добавлена обработка исключений при преобразовании строк в числа
- Улучшен пользовательский интерфейс с подсказками и сообщениями об ошибках
- Добавлена проверка границ массива при доступе к элементам

Основные трудности, возникшие в ходе работы:

- Организация корректного управления памятью в динамическом массиве
- Реализация эффективного алгоритма изменения размера массива
- Обеспечение безопасности при доступе к элементам массива
- Реализация проверки ввода для исключения ошибочных данных от пользователя
- Обработка исключений при преобразовании строк в числовые значения

В ходе работы были освоены навыки:

- Разработки классов в C++
- Работы с динамической памятью
- Реализации перегруженных операторов
- Создания консольных приложений с меню
- Тестирования программного обеспечения
- Обработки ошибок ввода и обеспечения надежности программы
- Использования механизма исключений для обработки ошибок