

**МИНОБРНАУКИ РОССИИ  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И.  
УЛЬЯНОВА (ЛЕНИНА)  
Кафедра Информационных систем**

**ОТЧЁТ  
по практической работе №1  
По дисциплине «Объектно-ориентированное программирование»**

Студент гр.  
Преподаватель

Шапошников А.А. 3376  
Егоров С.С.

Санкт-Петербург  
2024

## Задание

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

- создавать 0 массив (без элементов),
- создавать массив с N элементами и заполнять их заданными значениями,
- корректное уничтожение объектов класса,
- изменять размерность массива,
- вводить элементы массива из стандартного потока ввода (объект cin класса istream),
- выводить элементы массива в стандартный поток вывода (объект cout класса ostream),
- вычислять среднее и среднеквадратичное отклонение (СКО)

Создать консольное приложение согласно представленной на рис.1 диаграмме классов, предназначенное для работы с массивом **вещественных чисел**. Приложение должно включать основной модуль (функция main), модуль «application» и модуль «array».

В **основном модуле** консольного приложения (для языка C++ — это модуль с функцией main) должен создаваться объект класса "Консольное приложение" и вызываться его метод, который предоставляет пользователю **меню команд** приложения.

Модуль «**application**» должен содержать спецификацию класса "Консольное приложение" и реализацию его методов. Один из его методов должен выводить в консоль меню команд приложения, включающее:

- команду, иницирующую ввод с консоли значений, задающих объект массива (до ввода в программе должен быть задан массив по умолчанию);
- команду, иницирующую расчеты среднего и СКО элементов массива и вывод результатов расчета;
- команду, иницирующую сортировку элементов массива по возрастанию или убыванию;
- команду, иницирующую изменения размерности массива;
- команду, иницирующую изменения значения выбранного элемента массива;
- команду, иницирующую вывод в консоль элементов текущего объекта массива;
- команду выхода из приложения.

Модуль «**array**» должен содержать спецификацию класса "Массив" и реализацию его методов, необходимых для достижения цели разрабатываемого приложения. Описание класса должно использовать вместо типа `double` (вещественное число, заданное в условии) абстрактный тип ***number***, описание которого должно задаваться в отдельном заголовочном файле `number.h` с помощью оператора **`typedef double number`** (для C++).

Требуется реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и протестировать на них программу. Оформить отчет, сделать выводы по работе.

## СПЕЦИФИКАЦИИ КЛАССОВ

### Спецификации класса **CoreApp**:

Методы	
<code>private short mainMenu ()</code>	метод, отображающий меню и возвращающий выбранную опцию в виде числа
<code>private void waitForAction ()</code>	метод, ожидающий действий пользователя
<code>public CoreApp ()</code>	конструктор, создающий новый экземпляр класса <b>CoreApp</b>
<code>public int exec ()</code>	метод, запускающий цикл приложения и возвращающий код завершения

### Спецификации класса **Array**:

Атрибуты	
<code>private number* values</code>	указатель на массив чисел типа <b>number</b>
<code>private unsigned len</code>	количество элементов в массиве
Методы	
<code>public Array ()</code>	конструктор, который создает новый пустой экземпляр класса <b>Array</b>
<code>public Array (unsigned)</code>	конструктор, который создает новый экземпляр класса <b>Array</b> заданной длины
<code>public ~Array ()</code>	деструктор, который освобождает выделенную память для массива значений элементов

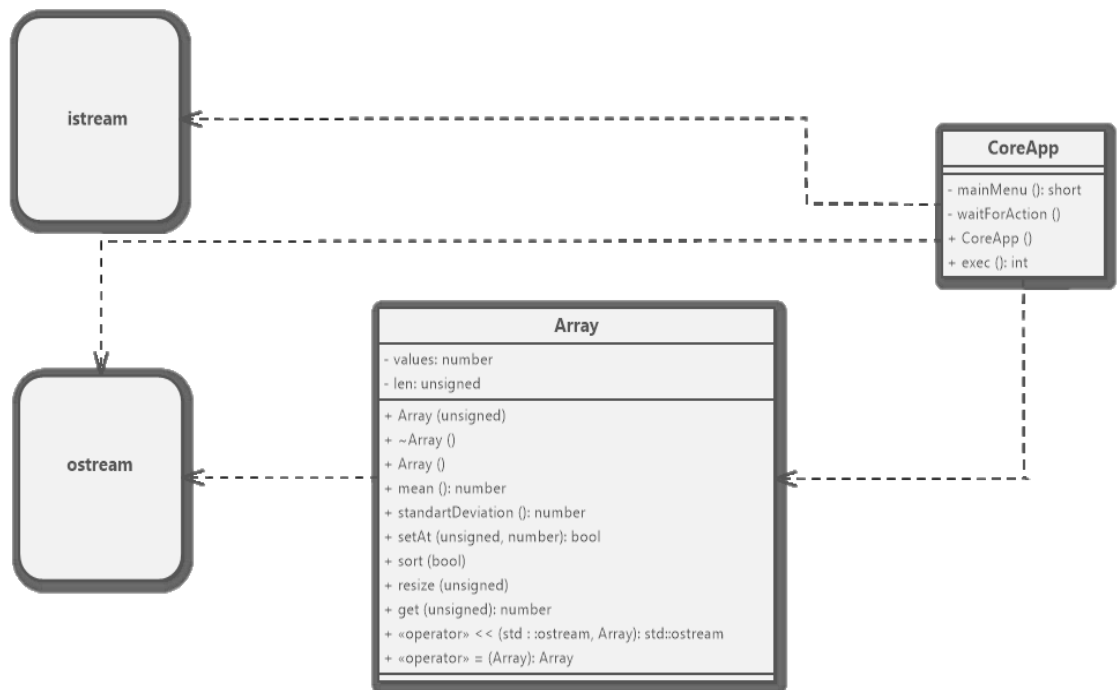
<code>public number mean ()</code>	метод, который возвращает среднее значение элементов в массиве
<code>public number standartDeviation ()</code>	метод, который возвращает стандартное отклонение элементов в массиве
<code>public bool setAt (unsigned, number)</code>	метод, который устанавливает значение элемента на заданной позиции
<code>public void sort(bool)</code>	метод, который сортирует массив элементов в порядке возрастания или убывания
<code>public number get(unsigned)</code>	метод, который возвращает значение элемента на заданной позиции
<code>friend std: ostream&amp; operator &lt;&lt;(std: ostream&amp;, const Array&amp;)</code>	перегрузка оператора вывода в поток, которая выводит значения элементов массива;
<code>public Array&amp; operator= (const Array&amp;)</code>	перегрузка оператора присваивания, которая копирует значения элементов из одного массива в другой.

Класс `Array` представляет собой динамический массив чисел типа `number`. Класс содержит методы для работы с элементами массива, такими как установка значения элемента по индексу, получение элемента по индексу, сортировка массива, изменение размеров массива, а также вычисление среднего значения и стандартного отклонения элементов массива. Класс также содержит перегруженный оператор вывода в поток, чтобы массив можно было легко вывести на экран.

Класс `CoreApp` представляет собой главный класс приложения, который содержит методы для управления его выполнением. Класс содержит главное меню приложения и метод для ожидания действия пользователя.

Выбор языка `C++` обусловлен его возможностями для работы с динамической памятью, перегрузкой операторов и поддержкой объектно-ориентированного программирования, также возможностью для написания кроссплатформенных программ и поддержкой многопоточности.

## ДИАГРАММА КЛАССОВ



## ПРИМЕРЫ РАБОТЫ ПРИЛОЖЕНИЯ

Заполнение массива вещественными числами:

```
1. Заполнение массива значениями
2. Расчёт СКО среднего
3. Сортировка массива
4. Изменить размер массива
5. Изменение элемента по индексу
6. Вывод элементов массива
0. Выход
>1
Размер массива: 5
1
2. 02
5. 45
7
9. 007 _
```

Расчёт среднего значения и СКО среднего:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>2
Среднее значение: 4.8954
СКО среднего: 3.35724
Press any key to continue . . . _
```

Вывод элементов массива в стандартный поток вывода:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>6
1 2.02 5.45 7 9.007 Press any key to continue . . .
```

Сортировка массива (в данном случае – по убыванию):

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>3
1.По возрастанию
2.По убыванию
>2
Успешно
Press any key to continue . . .
```

Проверяем результат:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>6
9.007 7 5.45 2.02 1 Press any key to continue . . . _
```

Изменение размера массива:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>4
Размер массива: 3
Press any key to continue . . .
```

Проверяем результат:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>6
9.007 7 5.45 Press any key to continue . . . _
```

Изменение элемента по индексу(индекс-0, значение-4):

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>5
Индекс элемента: 0
Значение: 4
Press any key to continue . . .
```

Проверяем результат:

```
1.Заполнение массива значениями
2.Расчёт СКО среднего
3.Сортировка массива
4.Изменить размер массива
5.Изменение элемента по индексу
6.Вывод элементов массива
0.Выход
>6
4 7 5.45 Press any key to continue . . .
```

## ВЫВОД

В данном задании было создано консольное приложение для работы с массивом вещественных чисел. Были разработаны три модуля – основной модуль, модуль приложения и модуль массива.

В основном модуле был создан объект класса "Консольное приложение" и вызван его метод, который предоставляет пользователю меню команд приложения.

Модуль "application" содержит спецификацию класса "Консольное приложение" и реализацию его методов. Один из его методов выводит в консоль меню команд приложения, включающее команды: иницилирующую ввод значений, иницилирующую расчеты, иницилирующую сортировку элементов, иницилирующую изменения размерности массива, иницилирующую изменения значения выбранного элемента массива и команду вывода элементов массива.

Модуль "array" содержит спецификацию класса "Массив" и реализацию его методов, необходимых для работы приложения. Описание класса



использует абстрактный тип `number`, заданный в отдельном заголовочном файле `number.h` с помощью оператора `typedef double number`.

Были использованы основы объектно-ориентированного программирования, такие как инициализация, вызов методов, присвоение переменных, разграничение доступа к полям и методам, скоупы классов, создание статических и динамических полей, констант, внутренних классов.

В результате работы было создано полноценное приложение для работы с массивом вещественных чисел, которое позволяет вводить, редактировать, рассчитывать статистические данные, сортировать и изменять размерность массива.