Федеральное государственное автономное образовательное учреждение

высшего образования «Санкт-Петербургский государственный

электротехнический университет «ЛЭТИ»

им. В.И. Ульянова (Ленина)»

**Кафедра информационных систем**

**ОТЧЕТ**

**по практической работе №1**

**по дисциплине «Объектно-ориентированное программирование»**

Студент гр. №3376 Михайлов Н.  
Студентка гр. №3376 Дегтярева М.  
  
Преподаватель Егоров С.С.

Санкт-Петербург  
2024

Задание на практическую работу:   
Создать консольное приложение предназначенное для работы с массивом вещественных чисел.  
Приложение должно включать основной модуль (функция main), модуль «application» и модуль «array». Приложение должно работать с разработанным классом “Массив” (Array) способным: создавать 0 массив, создавать массив с N элементами и заполнять их заданными значениями, корректно уничтожать объекты класса, изменять размерность массива, вводить элементы массива из istream, выводить элементы в ostream, вычислять среднее отклонение и СКО по формулам, сортировать элементы массива по возрастанию и убыванию.  
Так же Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

Таблица 1. Первичный протокол класса APPLICATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | | | | |
| идентификатор | тип |  | | семантическое описание |
| mainMenuItems | std::vector<std::string> | private | | Пункты основного меню |
| sortMenuItems | std::vector<std::string> | private | | Пункты для меню сортировки |
| arrow | std::string | private | | Стрелка |
| arrowPoss | unsigned | private | | Для отслеживания позиции стрелки |
| Методы | | | | |
| идентификатор | тип | | Область видимости | семантическое описание |
| Application(); | - | | public | Конструктор |
| ~Application() | - | | public | деструктор |
| MainMenu(); | Void | | public | Вывод основного меню |
| SortMenu(); | Void | | public | Вывод меню сортировки |
| GetArrowPoss() | unsigned | | public | Получение позиции стрелки |
| SetArrowPoss(unsigned t\_arrowPoss) | void | | public | Установка позиции стрелки |

Таблица 2. Первичный протокол класса ARRAY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | | | | |
| идентификатор | тип | область видимости | | семантическое описание |
| array | std::vector<number> | private | | Задающийся массив |
| Методы | | | | |
| идентификатор | тип | | область видимости | семантическое описание |
| Array() | - | | public | Конструктор |
| ~ Array() | - | | public | Деструктор |
| Create(); | bool | | public | Создание пустого массива |
| Sort(bool ascending = true); | bool | | public | Сортировка массива |
| SetValue(unsigned t\_index, number value); | bool | | public | Изменение значений элементов массива |
| ChangeSize(unsigned t\_size); | bool | | public | Изменение размера массива |
| Print(); | bool | | public | Вывод массива |
| Mean(); | number | | public | Расчет среднего арифметического значения |
| StandardDeviation(); | number | | public | Расчет СКО |
| GetArray() | std::vector<number> | | public | Получение массива |

Диаграмма классов программы:

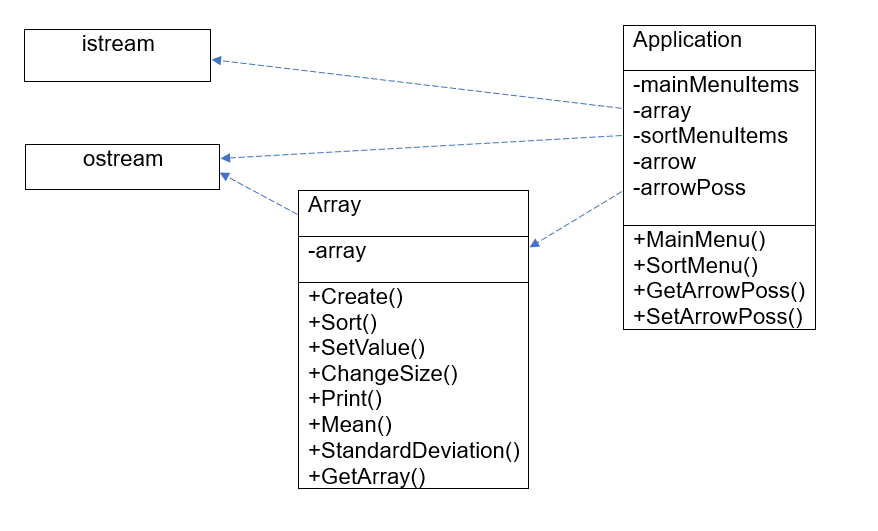


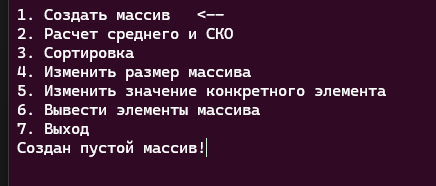
Рис.1. Диаграмма классов работы №1

Тесты работы программы:   
Контрольный пример.   
«-5; -3; 0; 16; 8; 3; 40; -10; 3,5; -3,7»

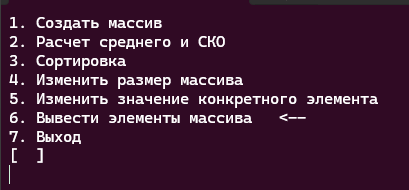
Проверим правильность работы программы на контрольном примере  
Расчетные результаты сортировки по возрастанию:  
«-10; -5; -3,7; -3; 0; 3; 3,5; 8; 16; 40»  
по убыванию:  
«40; 16; 8; 3,5; 3; 0; -3; -3,7; -5; -10»  
Расчетные результаты среднего значения и СКО:

|  |  |
| --- | --- |
|  | 4.9 |
| СКО | 13.568 |

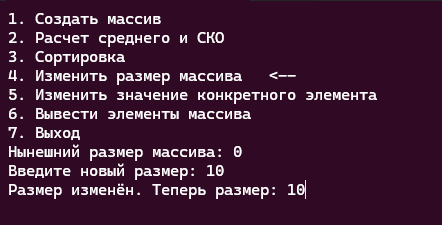
1. Создадим пустой массив



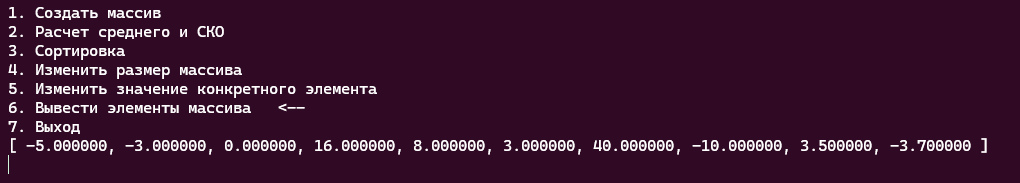
1. Проверим что массив создан и является пустым



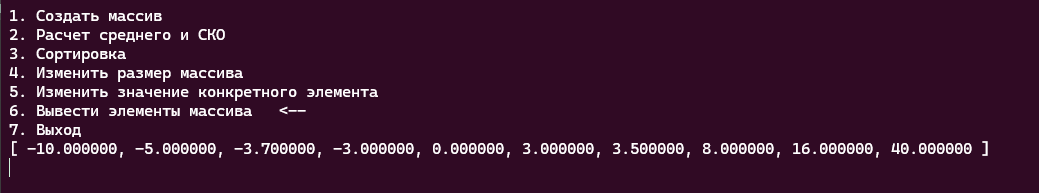
1. Изменим размер массива на нужный



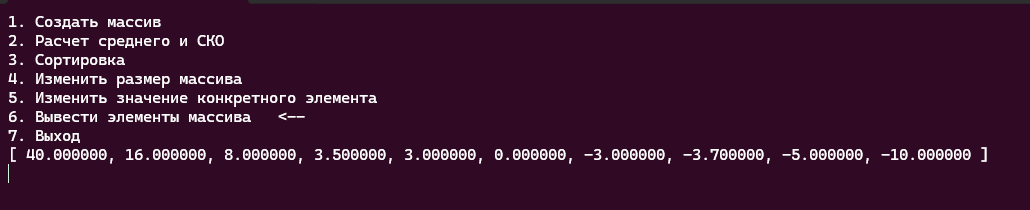
1. Изменим числа в массиве на числа из контрольного примера
2. Проверим массив



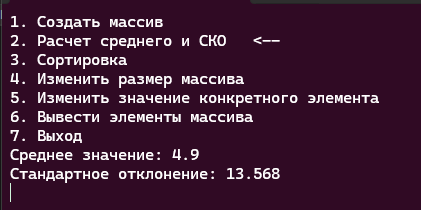
1. Отсортируем массив по возрастанию



1. Отсортируем массив по убыванию



1. Рассчитаем среднее значение и СКО по формуле, проверим совпадает ли с ожидаемым результатом.



Расчетные результаты совпадают с полученными. Программа работает корректно.

Выводы:   
В ходе работы было создано консольное приложение для работы с массивом вещественных чисел, способное: создавать массив с N элементами включая 0 и заполнять его заданными значениями, изменять размер массива, выводить все элементы массива. Для этого были созданы классы в соответствии с заданной диаграммой.   
В ходе тестирования ожидаемые результаты совпали с результатами работы программы.  
Программа написана согласно положениям и идеям парадигмы ООП.