

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

ПРАКТИЧЕСКАЯ РАБОТА №2
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: КОМПЛЕКСНЫЕ ЧИСЛА

Студенты гр. 4373
Преподаватель

Мельник А.А.
Шепелев Д.Н.
Егоров С.С.

Санкт-Петербург
2025

Задание на Практическую работу

Исходные данные: создать консольное приложение согласно представленной диаграмме классов, предназначенное для работы с массивом чисел. Для этого необходимо специфицировать пользовательские классы "Консольное приложение" и "Массив", т.е. задать атрибуты и методы указанных классов, а также распределить их по существующим областям видимости. Спецификация классов и реализация их методов должна обеспечивать реализацию отношений, указанных на диаграмме классов. В отчете представить аргументированное обоснование своего выбора.

семантическое описание

Спецификации классов

Таблица 1. Первичный протокол класса ConsoleApp

Атрибуты			
идентификатор	тип	область видимости	семантическое описание
countRows	int	private	Количество пунктов меню(длина массива menu)
menu	string*	private	Массив строк(пунктов меню)
Методы			
идентификатор	область видимости		семантическое описание
ConsoleApp	public		Конструктор
~ConsoleApp	public		Деструктор
startApp	public		Запуск приложения
choiseMenu	public		Меню выбора

Код первой практической претерпел изменения, так как вывод в консоль был переделан под другую библиотеку, совместимую с linux системами. Также была переписана часть класса ConsoleApp, для удобства разработки функционал функций createArr, calcAverenge, SortArr, changeSize, changeElem, printElem был пренесен в ChoiseMenu. Остальное осталось неизменным.

Таблица 2. Первичный протокол класса Array

Атрибуты			
идентификатор	тип	область видимости	семантическое описание
size	int	private	Размер массива
arr	number*	private	Указатель на первый элемент массива
Методы			
идентификатор	область видимости		семантическое описание
Array	public		Конструктор(по умолчанию)
Array	public		Конструктор(с вводом размера и элементов)
~Array	public		Деструктор
getSize	public		Получение размера массива
changeSize	public		Изменение размерности
cinNumber	public		Ввод числа из потока ввода
coutNumber	public		Вывод числа в потока вывода
calcAverenge	public		Расчёт среднего
calcStandardDeviation	public		Расчёт СКО
sort	public		Сортировка по убыванию/возрастанию

Таблица 3. Первичный протокол класса Tcomplex

Атрибуты			
идентификатор	тип	область видимости	семантическое описание
real	double	private	Вещественная часть комплексного числа
imag	double	private	Мнимая часть комплексного числа
Методы			
идентификатор	область видимости		семантическое описание
TComplex	public		Конструктор(по умолчанию) (= 0)
TComplex	public		Конструктор вещественного числа
TComplex	public		Конструктор комплексного числа
getReal	public		Получение вещественной части
getImag	public		Получение мнимой части
putReal	public		Изменение вещественной части
putImag	public		Изменение мнимой части
printw	public		Перегрузка вывода
scanw	public		Перегрузка ввода
sqrt	public		Корень из комплексного числа

Диаграмма классов

На рисунке 1 представлена диаграмма классов, дополненная атрибутами и методами.

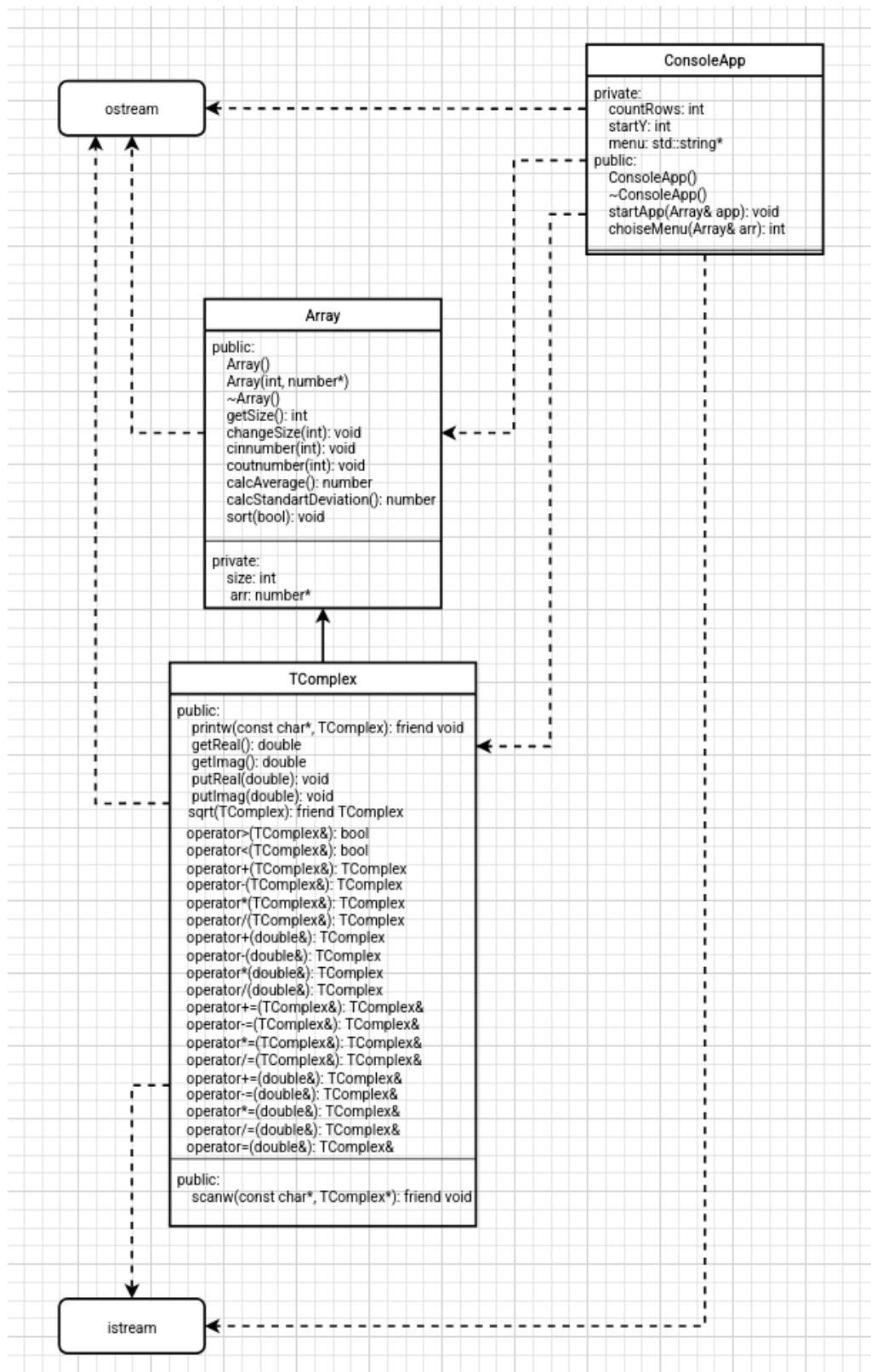


Рис 1 Диаграмма классов

Описание контрольного примера с исходными и ожидаемыми расчетными данными

1. Задаётся массив. Размерность массива — 5. Вводятся значения элементов:
(1.5, -2.3), (0, 4.7), (-3.14, 0), (2.71, 1.41), (-5.5, -3.3)
2. Рассчитывается среднее значение и СКО по формулам: $-0.886+0.102i$;
 $5.804609+5.014419i$
3. Массив сортируется в порядке возрастания: (1.5, -2.3), (2.71, 1.41), (-3.14, 0), (0, 4.7), (-5.5, -3.3); и в порядке убывания: (-5.5, -3.3), (0, 4.7), (-3.14, 0), (2.71, 1.41), (1.5, -2.3)
4. Размерность массива изменяется на 4. Новый массив: (-5.5, -3.3), (0, 4.7), (-3.14, 0), (2.71, 1.41)
5. Элемент под индексом 2: (-3.14, 0), - изменяется на (1.72, -2.91). Новый массив: (-5.5, -3.3), (0, 4.7), (1.72, -2.91), (2.71, 1.41)
6. Выводим элемент с индексом 3: (2.71, 1.41)
7. Выходим из программы

СКРИНШОТЫ РАБОТЫ ПРОГРАММЫ НА КОНТРОЛЬНЫХ ПРИМЕРАХ

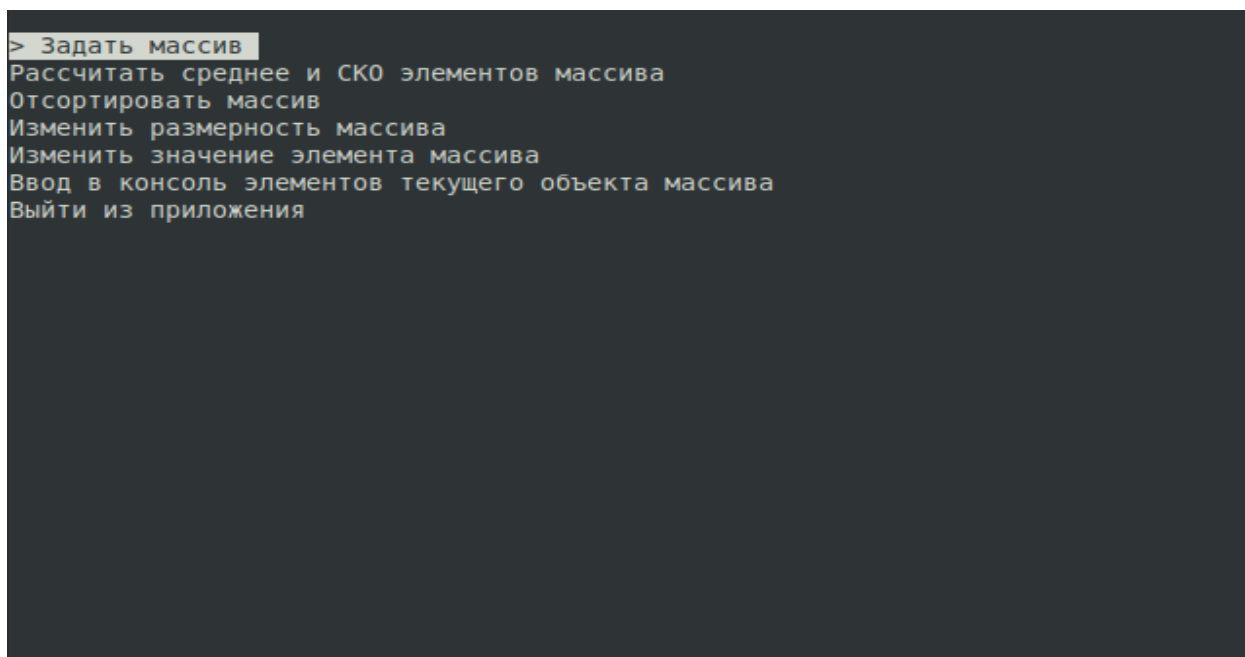


Рис 2. Запускаем программу

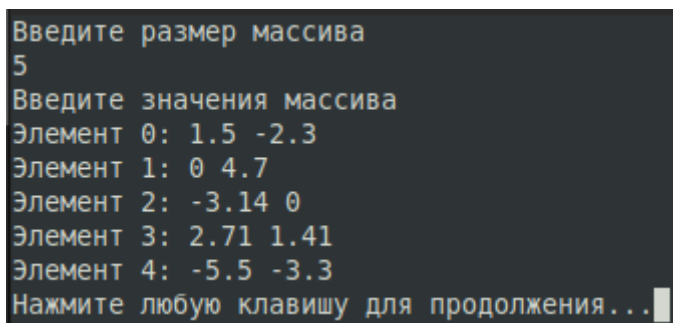


Рис 3. Вводим числа и выводим массив

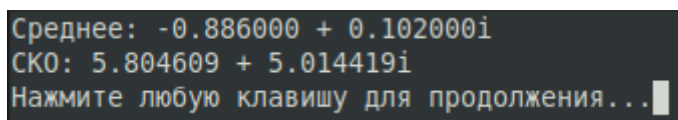


Рис 4. Вычисляем среднее и СКО

```
Отсортировать массив в порядке возрастания? (Отказ для убывания) (Y/n): Y
Отсортированный массив:
1.500000 - 2.300000i 2.710000 + 1.410000i -3.140000 + 0.000000i 0.000000 + 4.700000i -5.500000 - 3.300000i
Нажмите любую клавишу для продолжения...
```

Рис 5. Сортируем массив в порядке возрастания

```
Отсортировать массив в порядке возрастания? (Отказ для убывания) (Y/n): N
Отсортированный массив:
-5.500000 - 3.300000i 0.000000 + 4.700000i -3.140000 + 0.000000i 2.710000 + 1.410000i 1.500000 - 2.300000i
Нажмите любую клавишу для продолжения...
```

Рис 6. Сортируем массив в порядке убывания

```
Введите новый размер массива
4
Новый массив: -5.500000 - 3.300000i 0.000000 + 4.700000i -3.140000 + 0.000000i 2.710000 + 1.410000i
Нажмите любую клавишу для продолжения...
```

Рис 7. Изменяем размерность массива

```
Введите номер элемента для изменения (0-3): 2
Текущее значение: -3.140000 + 0.000000i
Введите новое значение: 1.72 -2.91
Элемент изменен.
Нажмите любую клавишу для продолжения...
```

Рис 8. Изменяем элемент под индексом 2.

```
Введите номер элемента (0-3): 3
Элемент 3: 2.710000 + 1.410000i
Нажмите любую клавишу для продолжения...
```

Рис 9. Выводим элемент с индексом 3.

ВЫВОДЫ ПО ВЫПОЛНЕНИЮ РАБОТЫ

В рамках данной практической работы была дополнена программа, реализованная в практической работе №1. Был создан класс TComplex, реализующий представление и методы работы с комплексными числами. Также меню программы было переписано с использованием библиотеки ncurses для возможности работы с программой на Linux.