

## **ВВЕДЕНИЕ**

Цель – разработка программы, которая удаляет все дубликаты из массива целых чисел. Программа должна корректно обрабатывать массивы любого размера и выводить итоговый массив без повторяющихся элементов.

Задачи:

- Составить IDEF0-диаграмму проекта;
- Составить блок-схемы алгоритмов;
- Реализовать алгоритмы на языке Object Pascal;
- Протестировать реализации алгоритмов.

## Аналитическая часть

Структура представлена на рисунке 1 в виде IDEF0-нотации. Задача разделяется на несколько блоков, представленных на рисунке 2:

- Ввод данных с клавиатуры;
- Инициализация матрицы ;
- Вставка элемента в матрицу;
- Вывод данных на экран.



Рисунок 1

– Общая IDEF0-нотация

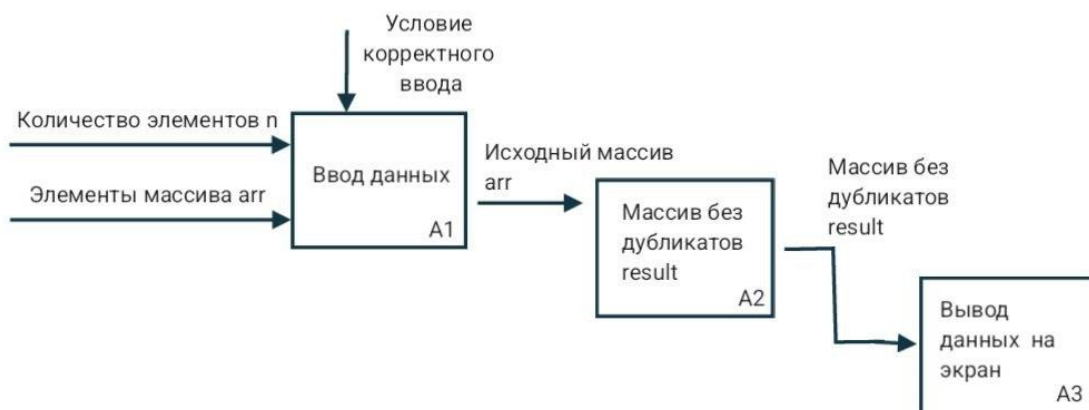


Рисунок 2 – Подробная IDEF0-нотация

## Конструкторская часть

Блоки IDEF0-диаграммы представляют собой 1-2 действия, ввиду чего рациональнее отобразить алгоритм всего проекта целиком без разбиения каждого блока на отдельные процедуры.

Блок-схема алгоритма представлена на рисунке 1. Элементы, отвечающие за интерфейс пользователя, на блок-схеме не отображены; текстовые сообщения, ввиду малозначимости их дословного приведения, представлены сокращенно.

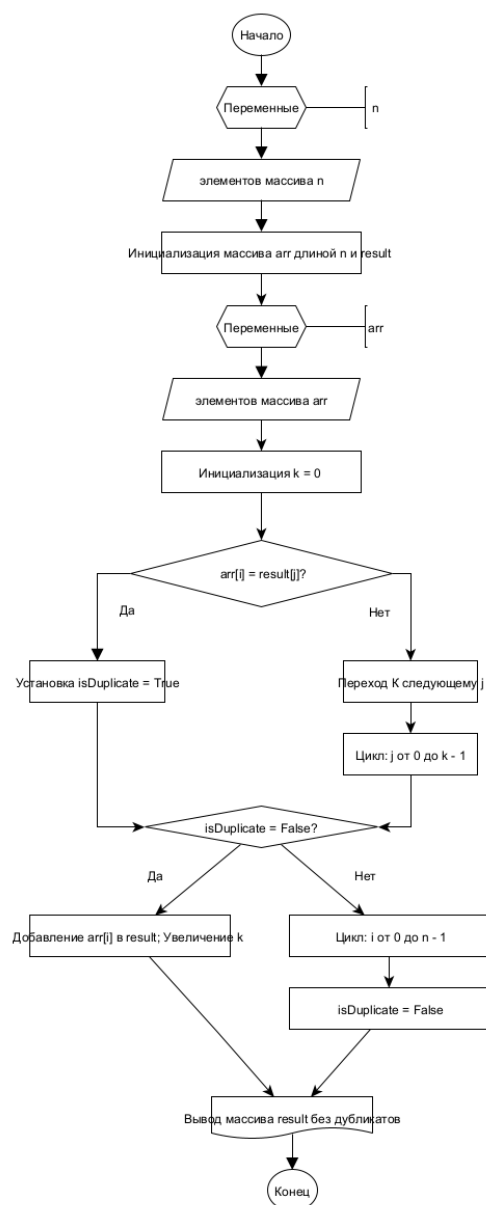


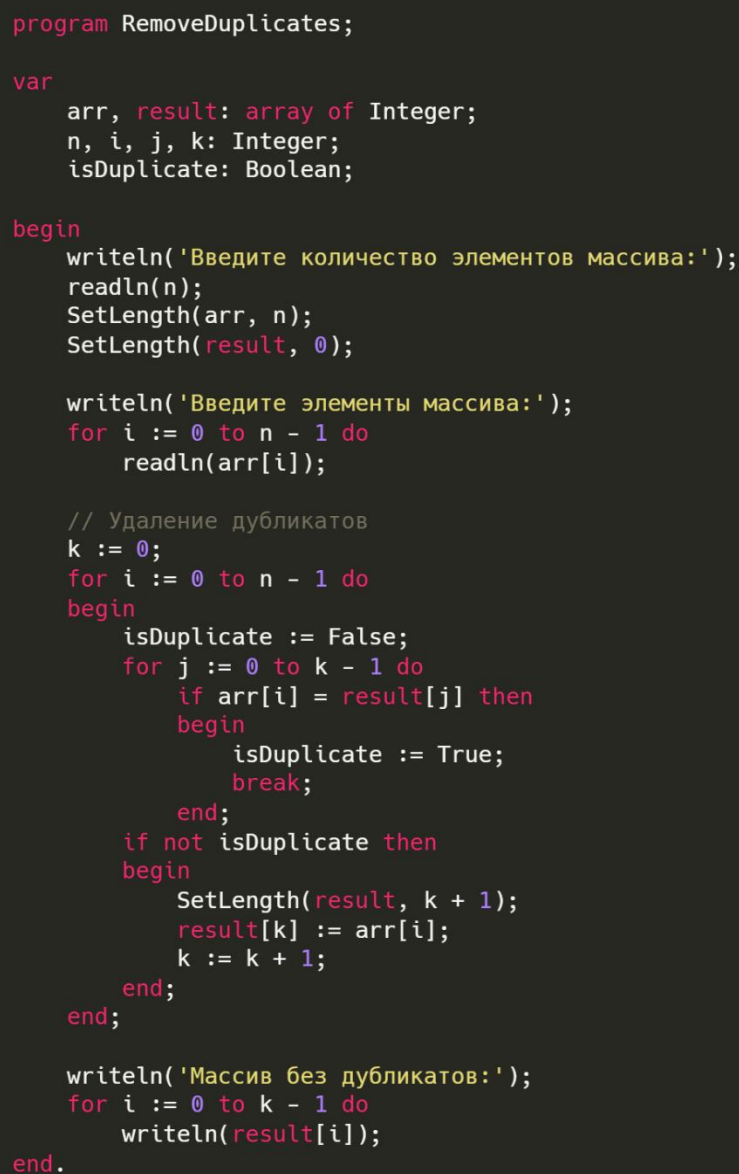
Рисунок 1 – Блок-схема алгоритма программы

## Технологическая часть

### Реализация алгоритма

В настоящем разделе представлена реализация алгоритма, чья блок-схема представлена на рисунке 1. Реализация была произведена с помощью языка программирования Pascal и представлена в листинге 1.

Листинг 1 – Программа алгоритма



```
program RemoveDuplicates;

var
  arr, result: array of Integer;
  n, i, j, k: Integer;
  isDuplicate: Boolean;

begin
  writeln('Введите количество элементов массива:');
  readln(n);
  SetLength(arr, n);
  SetLength(result, 0);

  writeln('Введите элементы массива:');
  for i := 0 to n - 1 do
    readln(arr[i]);

  // Удаление дубликатов
  k := 0;
  for i := 0 to n - 1 do
    begin
      isDuplicate := False;
      for j := 0 to k - 1 do
        if arr[i] = result[j] then
          begin
            isDuplicate := True;
            break;
          end;
      if not isDuplicate then
        begin
          SetLength(result, k + 1);
          result[k] := arr[i];
          k := k + 1;
        end;
    end;

  writeln('Массив без дубликатов:');
  for i := 0 to k - 1 do
    writeln(result[i]);
end.
```

## Тестирование реализации

Для данной реализации были использованы методы тестирования черным ящиком – метод эквивалентного разбиения – и белым ящиком – метод комбинаторного покрытия условий и решений.

Были выделены следующие эквивалентные классы возможных решений:

1. Введен массив с дубликатами, программа корректно удаляет дубликаты и выводит итоговый массив.
2. Введен массив без дубликатов, программа выводит массив без изменений.
3. Введен пустой массив, программа выводит пустой результат.
4. Введены некорректные или нечисловые данные, программа выводит системную ошибку.

Результаты тестов представлены в таблице 1. Сообщения о системной ошибке предусмотрены для некорректного ввода данных.

Таблица 1 – Тесты реализации программы

№	Входные данные	Выходные данные
1	1, 2, 2, 3, 4, 4, 5	1, 2, 3, 4, 5
2	5, 5, 5, 5, 5	5
3	10, 20, 10, 30, 20	10, 20, 30
4	'1, 1, 1, 1, 1	1
5	пустой массив	пустой массив
6	"abc", 1, 2	Системная ошибка (введено не число)

## **ЗАКЛЮЧЕНИЕ**

В настоящей работе была составлена программа для удаления дубликатов из массива целых чисел.

Задачи:

- Составлена IDEF0-диаграмма проекта;
- Составлены блок-схемы алгоритмов;
- Алгоритмы реализованы на языке Object Pascal;
- Реализации алгоритмов протестированы, были подобраны классы ошибок.