

ВВЕДЕНИЕ

Цель – данной работы – разработать программу, которая выполняет следующие операции с массивом целых чисел.

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;
- Заполнение массива случайными числами;
- Сортировка массива по убыванию;
- Подсчет уникальных элементов;
- Разделение на четные и нечетные элементы.
- Вывод данных на экран.

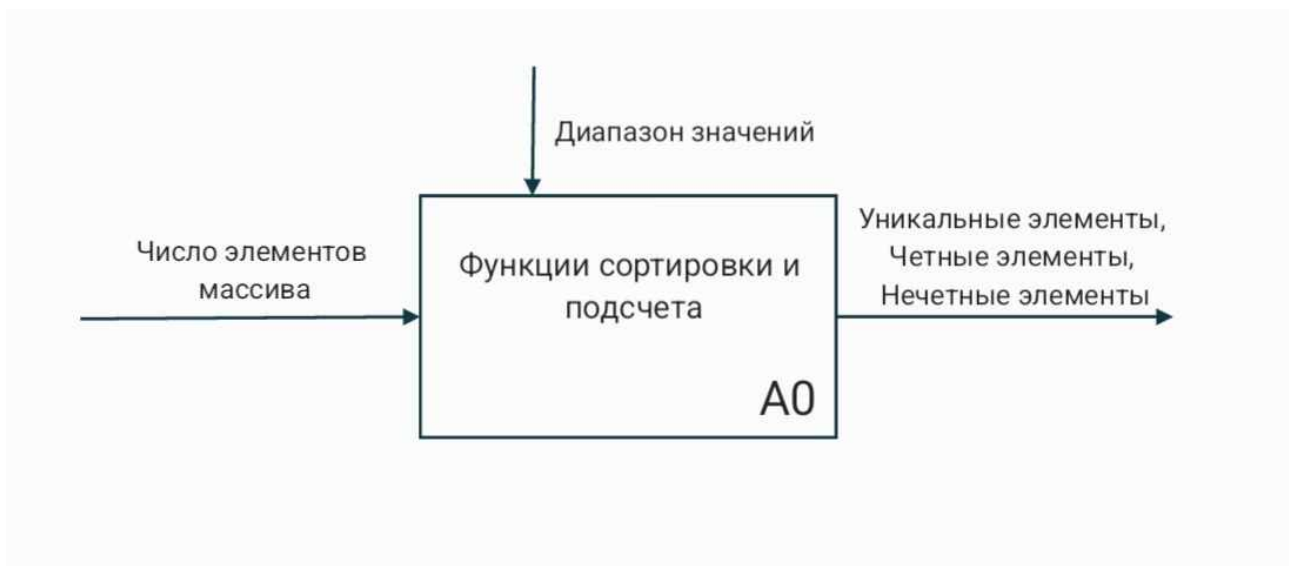


Рисунок 1 – Общая IDEF0-нотация

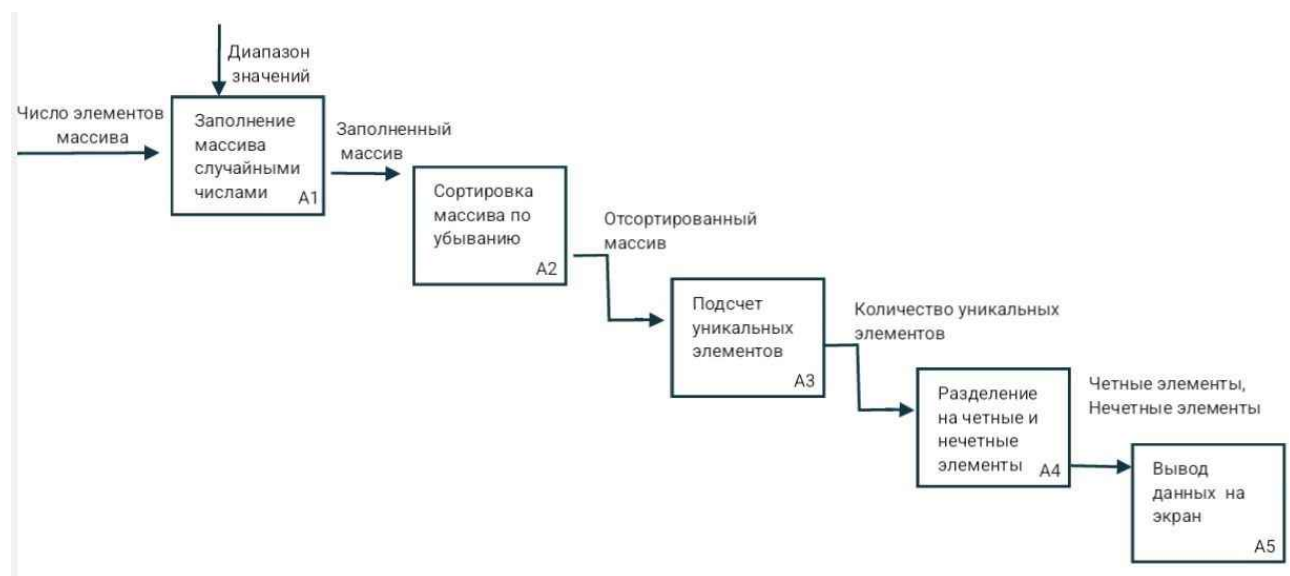


Рисунок 2 – Подробная IDEF0-нотация

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.

```

program ArrayOperations;

uses SysUtils;

var
  arr: array of Integer;
  n, i, uniqueCount: Integer;
  evenArr, oddArr: string;

function CountUniqueElements(a: array of Integer): Integer;
var
  i, count: Integer;
begin
  count := 1;
  for i := 1 to High(a) do
    if a[i] <> a[i - 1] then
      count := count + 1;
  CountUniqueElements := count;
end;

procedure SortDescending(var a: array of Integer);
var
  i, j, temp: Integer;
begin
  for i := 0 to High(a) - 1 do
    for j := i + 1 to High(a) do
      if a[i] < a[j] then
        begin
          temp := a[i];
          a[i] := a[j];
          a[j] := temp;
        end;
  end;

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

  // Заполнение массива случайными числами от 1 до 20
  for i := 0 to n - 1 do
    arr[i] := Random(20) + 1;

  // Сортировка массива по убыванию
  SortDescending(arr);

  // Подсчет количества уникальных элементов
  uniqueCount := CountUniqueElements(arr);
  writeln('Количество уникальных элементов: ', uniqueCount);

  // Разделение на четные и нечетные элементы
  evenArr := '';
  oddArr := '';
  for i := 0 to High(arr) do
    if arr[i] mod 2 = 0 then
      evenArr := evenArr + IntToStr(arr[i]) + ' '
    else
      oddArr := oddArr + IntToStr(arr[i]) + ' ';

  // Вывод четных и нечетных элементов
  writeln('Четные элементы: ', evenArr);
  writeln('Нечетные элементы: ', oddArr);
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. Массив заполнен случайными числами от 1 до 20.
2. Отсортированный массив, содержит как четные, так и нечетные числа.
3. Массив, полностью состоящий из одинаковых чисел.
4. Некорректное значение для размера массива.

Результаты тестов представлены в таблице 1.

Таблица 1 – Тесты реализации программы

| № | Входные данные | Выходные данные |
|---|--|--|
| 1 | n=10, элементы: 3, 5, 2, 20, 8, 10, 1, 15, 20, 2 | Отсортированный массив: 20, 20, 15, 10, 8, 5, 3, 2, 2, 1 Четные: 20, 20, 10, 8, 2, 2 Нечетные: 15, 5, 3, 1 Уникальные элементы: 7 |
| 2 | n=5, элементы: 1, 1, 1, 1, 1 | Отсортированный массив : 1, 1, 1, 1, 1 Четные: - Нечетные: 1, 1, 1, 1, 1 Уникальные элементы: 1 |
| 3 | n=8, элементы: 4, 6, 8, 10, 12, 14, 16, 18 | Отсортированный массив : 18, 16, 14, 12, 10, 8, 6, 4 Четные: 18, 16, 14, 12, 10, 8, 6, 4 Нечетные: - Уникальные элементы: 8 |
| 4 | n=-15 | Системная ошибка |
| 5 | n='abc' | Сообщение «Произведения одинаковые» |

ЗАКЛЮЧЕНИЕ

В настоящей работе была составлена программа для создания массива, сортировки по убыванию, подсчета уникальных значений и вывода четных и нечетных элементов.

Задачи:

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