

# Введение

Цель - Создать программу на языке Pascal, которая принимает строку от пользователя и удаляет из нее все пробелы, чтобы получить строку без пробелов. Это может быть полезно для обработки текстовых данных, где пробелы не требуются (например, при очистке данных перед анализом или обработкой).

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;
- Создание массива;
- Выполнение процедур;
- Вывод матрицы на экран.



Рисунок 1 - Общая IDEF0-Нотация

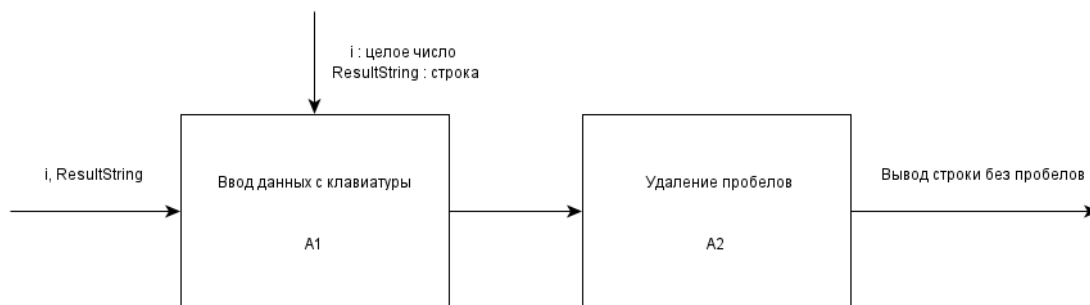


Рисунок 2 - Подробная IDEF0-Нотация

## ■ 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.

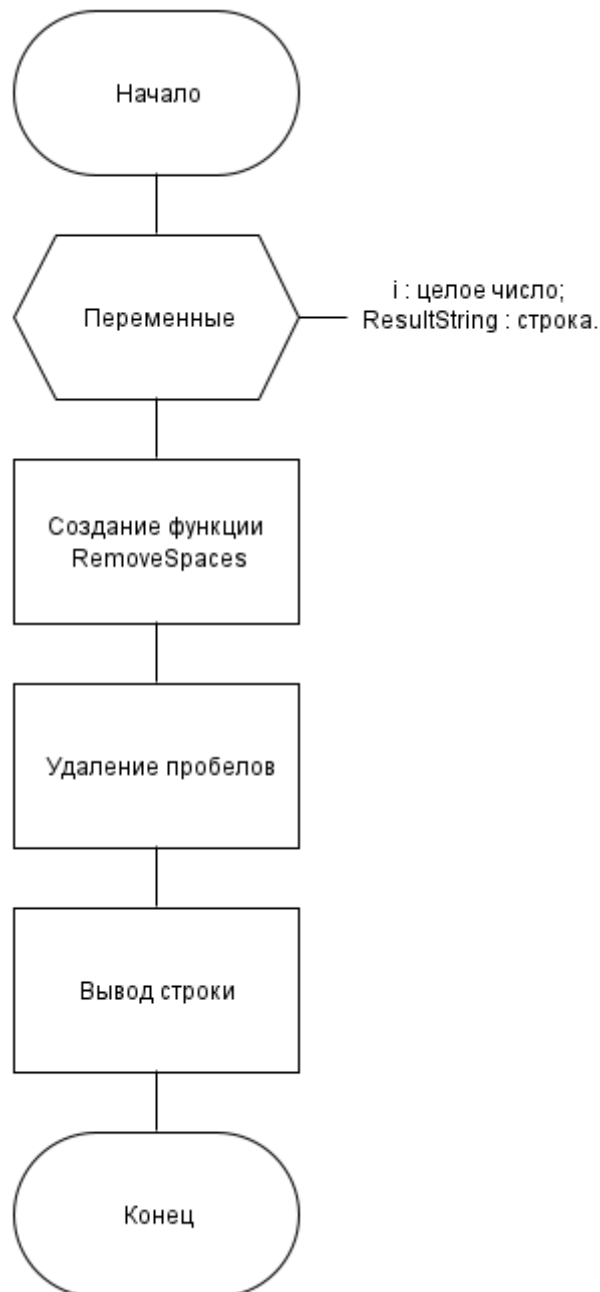


Рисунок 1 - Блок-схема алгоритма программы

## ■ 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 RemoveSpaces;

uses
  SysUtils;

function RemoveSpaces(const Input: string): string;
var
  i: Integer;
  ResultString: string;
begin
  ResultString := '';
  for i := 1 to Length(Input) do
    begin
      if Input[i] <> ' ' then
        ResultString := ResultString + Input[i];
    end;
  RemoveSpaces := ResultString;
end;

var
  Original, Modified: string;

begin
  Write('Введите строку: ');
  ReadLn(Original);
  Modified := RemoveSpaces(Original);
  WriteLn('Строка без пробелов: ', Modified);
end.
```

---

# Тестирование реализации

Для тестирования программы применим:

## 1. Метод эквивалентного разбиения:

- Делим входные данные на классы, которые покрывают различные типы ввода:
  - Пустая строка.
  - Строка без пробелов.
  - Строка, содержащая только пробелы.
  - Строка с пробелами в начале, середине и конце.
  - Строка с комбинацией символов (буквы, цифры, специальные символы) и пробелами.

## 2. Метод комбинаторного покрытия условий и решений:

- Проверяем все комбинации, в которых пробелы встречаются в разных частях строки (в начале, конце, середине).
- Учитываем различные типы символов, соседствующих с пробелами.

## Результаты тестирования

Тест №	Входные данные	Ожидаемый результат
1	''	''
2	'HelloWorld'	'HelloWorld'
3	' '	''
4	' Hello World '	'HelloWorld'
5	'Pascal 123 '	'Pascal123'
6	' a b c '	'abc'
7	'Symbols ! @ # \$ '	'Symbols!@#'\$
8	'123 456 789 '	'123456789'
9	'NoSpacesHere'	'NoSpacesHere'
10	' Mixed case '	'Mixedcase'

## Заключение

В настоящей работе была составлена программа для работы с удалением пробелов в строке.

Задачи :

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