

## **ВВЕДЕНИЕ**

Цель – разработка функции для сравнения двух строк без учета регистра. Программа должна корректно обрабатывать строки любого размера и возвращать результат сравнения: равны ли строки или нет.

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;
- Преобразование строк ;
- Сравнение строк;
- 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

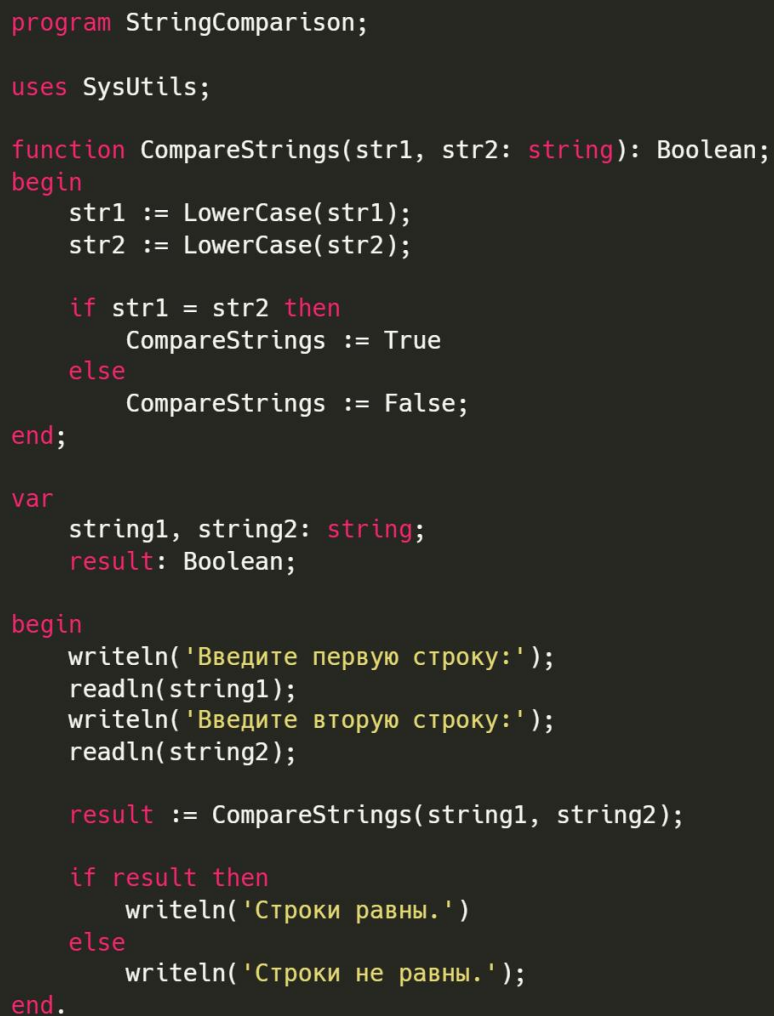


## Технологическая часть

### 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

The image shows a screenshot of a Pascal program in a code editor. The editor has a dark background with light-colored text. At the top left of the code area, there are three colored circles: red, yellow, and green. The code is written in Pascal and implements a string comparison algorithm. It defines a function 'CompareStrings' that takes two strings and returns a Boolean value. The main program uses this function to compare two strings entered by the user and prints the result.

```
program StringComparison;  
  
uses SysUtils;  
  
function CompareStrings(str1, str2: string): Boolean;  
begin  
    str1 := LowerCase(str1);  
    str2 := LowerCase(str2);  
  
    if str1 = str2 then  
        CompareStrings := True  
    else  
        CompareStrings := False;  
end;  
  
var  
    string1, string2: string;  
    result: Boolean;  
  
begin  
    writeln('Введите первую строку:');  
    readln(string1);  
    writeln('Введите вторую строку:');  
    readln(string2);  
  
    result := CompareStrings(string1, string2);  
  
    if result then  
        writeln('Строки равны.')  
    else  
        writeln('Строки не равны.');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. Строки полностью совпадают (включая регистр).
2. Строки совпадают, но регистр отличается.
3. Строки не совпадают.
4. Одна или обе строки пусты.
5. Ввод некорректных данных, отличных от строки.

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	Первая строка: "hello", Вторая: "hello"	Строки равны
2	Строки равны	Строки равны
3	Первая строка: "Pascal", Вторая: "Python"	Строки не равны
4	Первая строка: "", Вторая: ""	Строки равны
5	Первая строка: "123ABC", Вторая: "123abc"	Строки равны
6	Первая строка: "Hello", Вторая: "World"	Строки не равны

## **ЗАКЛЮЧЕНИЕ**

В настоящей работе была составлена программа для сравнения двух строк без учета регистра. Программа успешно протестирована и корректно выполняет заданные функции, обеспечивая правильное сравнение строк.

Задачи:

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