

Практическое занятие 3. Отладка программы обработки данных

Предлагаемая вашему вниманию консольная программа PassStatsBasic предназначена для обработки результатов тестирования персонала. Входными данными является файл marksheet.csv (текстовый файл значений, разделённых запятыми). В каждой записи файла содержится символ, обозначающий пол сотрудника, М – мужской, F – женский, разделитель – запятую, и число, обозначающее результат прохождения теста в диапазоне от 0 до 100. Например:

M,50

F.45

M,77

Если результат больше, чем заданная в программе константа PASSMARK = 40, сотрудник считается прошедшим тест. Программа подсчитывает процент прошедших тест отдельно для мужчин и для женщин.

```

program PassStatsBasic;
{$APPTYPE CONSOLE}
//Reads scores for males and females from a csv file
//and outputs pass rate for each.

uses
  SysUtils, Strutils;
const
  FILE_NAME = 'marksheet.csv';
  PASSMARK = 40;
var
  intCurrentScore, intMales, intFemales, intMalePasses, intFemalePasses : integer;
  strCurrentScore, strMalePassRate, strFemalepassRate : string;
  charCurrentGender : char;
  rMalePassRate, rFemalePassRate : real;

procedure CalcStats(FileName : string);
var
  Marks : Text;
  strCurrentLine : string;
  ErrorCode : integer;
begin
  intMales := 0;
  intFemales := 0;
  intMalePasses := 0;
  intFemalepasses := 0;
  intCurrentScore := 0;
  assignFile(Marks, FileName);
  reset(Marks);
  while not eof(Marks) do
    begin
      readln(Marks, strCurrentLine);
      charCurrentGender := LeftStr(strCurrentLine, 1)[1];
      charCurrentGender := UpCase(charCurrentGender);
      strCurrentScore := rightStr(strCurrentLine, length(strCurrentLine) - val(strCurrentScore, intCurrentScore, ErrorCode));
      if charCurrentGender = 'M' then
        begin
          inc(intMales);
          if intCurrentScore >= PASSMARK then
            inc(intMalePasses);
        end;
    end;
end;

```

```

        inc(intMalePasses);
    end
else //females
begin
    inc(intFemales);
    if intCurrentscore >= PASSMARK then
        inc(intFemalePasses);
    end;
end;
closeFile(Marks);
if intMales = 0 then
begin
    writeln('No males');
    strMalePassRate := 'NoMales';
end
else
begin
    rMalePassRate := intMalePasses * 100 / intMales;
    strMalePassRate := FloatToStrf(rMalePassRate, ffFixed, 6, 2);
end;
if intFemales = 0 then
begin
    writeln('No females');
    strFemalePassRate := 'NoFemales';
end
else
begin
    rFemalePassRate := intFemalePasses * 100 / intFemales;
    strFemalePassRate := FloatToStrf(rFemalePassRate, ffFixed, 6, 2);
end;
writeln('Male Pass Rate (%): ', strMalePassRate);
writeln('Female Pass Rate (%): ', strFemalePassRate);
end;

begin
    CalcStats(FILE_NAME);
    readln;
end.

```

Листинг 1.

Данная программа не отвечает требованиям заказчика, поскольку не имеет средств диагностики и выдачи сообщений, способных обнаружить ошибки в исходных данных, таких как: отсутствие файла исходных данных, отсутствие данных в файле, нарушение структуры данных, недопустимые символы в данных, отсутствие запятой, выход числа за пределы диапазона. Выдаваемое диагностическое сообщение должно содержать информацию о характере ошибки и номер строки файла исходных данных.

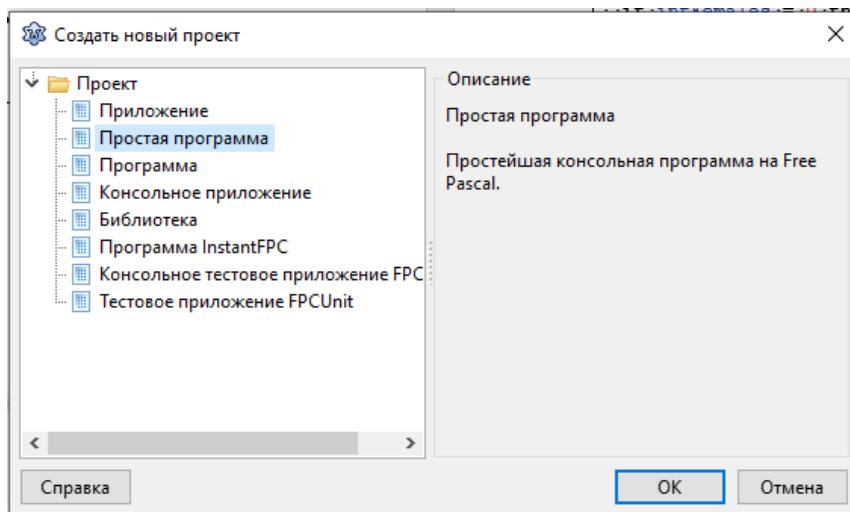
В то же время, программа должна правильно рассчитывать проценты прошедших, не учитывая пустые строки и строки, содержащие одну запятую.

Задание:

Отладить и доработать программу обработки статистических данных. Добавит в программу операторы обнаружения ошибок исходных данных и выдачи диагностических сообщений.

Ход выполнения работы:

1. Создать новый проект Lazarus, как простую программу, сохранить его в новой папке под именем `PassStats<ВАША ФАМИЛИЯ>` (содержащем Вашу фамилию, написанную латиницей)



2. Скопировать программу (Листинг 1), в файле модуля проекта заменить имеющуюся там заготовку скопированным текстом программы.
3. Проанализировать логику работы программы.
4. Внести необходимые изменения.
5. Оформит отчёт о выполненной работе, сдать на проверку файл отчёта и архив с папкой проекта без исполняемого модуля.