МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Частное учреждение образования

«Гродненский колледж бизнеса и права»

**Лабораторная работа № 20**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

**Тема:** Решение статистических задач

для учащихся 2 курса специальности

2-40 01 01 «Программное обеспечение информационных технологий»

**ЛАБОРАТОРНАЯ РАБОТА № 20**

Тема: Решение статистических задач.

Цель:

Образовательная**:**

* Обучить основным алгоритмам обхода графа и научиться решать задачи обхода графа на основе поиска в ширину и поиска в глубину,

Развивающая:

* научить анализировать алгоритмы обхода графа и научить решать задачи обхода графа на основе поиска в ширину и поиска в глубину,
* создать условия для развития способности четко формулировать свои мысли.

Воспитательная:

* воспитывать в обучающихся средствами урока уверенность в своих силах;

воспитывать сознательное и серьёзного отношения обучающихся к учебной дисциплине, убеждая их в том, что полученные знания пригодятся им в будущей деятельности.

Задачи: Освоение теоретического материала и выполнение индивидуального задания.

**ЗАДАЧИ**

Условие: Решение системы линейных алгебраических уравнений методом простых итераций Якоби.

Контрольный пример. Решить систему уравнений с точностью = 10-5.

-3.0x1 + 0.5x2 + 0.5x3 = -56.65

0.5x1 - 6.0x2 + 0.5x3 = -160

0.5x1 + 0.5x2 - 3.0x3 = -210

Алгоритм: Предоставлен преподавателю в письменном виде.

Решение:

**type**

vector = **array**[1..3] **of** real;

matrix = **array**[1..3, 1..3] **of** real;

**const**

n: integer = 3;

e: real = power(10, -5);

**var**

i, j, k, max: integer;

Ek, sum1, sum2: real;

a, alpha: matrix;

b, beta, x\_curr, x\_prev, x\_tmp: vector;

**begin**

a[1, 1] := -3; a[1, 2] := 0.5; a[1, 3] := 0.5; b[1] := -56.65;

a[2, 1] := 0.5; a[2, 2] := -6; a[2, 3] := 0.5; b[2] := -160;

a[3, 1] := 0.5; a[3, 2] := 0.5; a[3, 3] := -3; b[3] := -210;

Writeln('Исходная матрица имеет вид: ');

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

write(' ', a[i, j]:1:3);

writeln;

**end**;

Writeln('Преобразованная матрица имеет вид: ');

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

**if** i <> j **then**

alpha[i, j] := (-a[i, j]) / (a[i, i])

**else**

alpha[i, j] := 0;

**end**;

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

write(' ', alpha[i, j]:1:3);

writeln;

**end**;

Writeln('Проверим достаточное условие сходимости итерационной последовательности: ' );

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

Ek := Ek + sqr(alpha[i, j]);

**end**;

Ek := sqrt(Ek);

**if** Ek > 1 **then**

**begin**

writeln(' Eвклидова норма матрицы ', Ek:1:3, ' > 1. Условие не выполнено.');

**exit**;

**end**

**else**

**begin**

writeln(' Eвклидова норма матрицы ', Ek:1:3, ' < 1. Условие выполнено.');

**end**;

Ek := 0;

sum1 := 0;

sum2 := 0;

k := 0;

**for** i := 1 **to** n **do**

beta[i] := b[i] / a[i, i];

**repeat**

inc(k);

**if** k <> 1 **then**

**for** i := 1 **to** n **do**

x\_prev[i] := x\_tmp[i];

**for** i := 1 **to** n **do**

**begin**

sum1 := 0;

**for** j := 1 **to** i - 1 **do**

sum1 := sum1 + a[i, j] \* x\_prev[j];

sum2 := 0;

**for** j := i + 1 **to** n **do**

sum2 := sum2 + a[i, j] \* x\_prev[j];

x\_curr[i] := (b[i] - sum1 - sum2) / a[i, i];

**end**;

**for** i := 1 **to** n **do**

x\_tmp[i] := x\_curr[i];

max := 1;

**for** i := 1 **to** n - 1 **do**

**if** abs(x\_curr[max] - x\_prev[max]) < abs(x\_curr[i + 1] - x\_prev[i + 1]) **then**

max := i + 1;

**until** abs(x\_curr[max] - x\_prev[max]) < e;

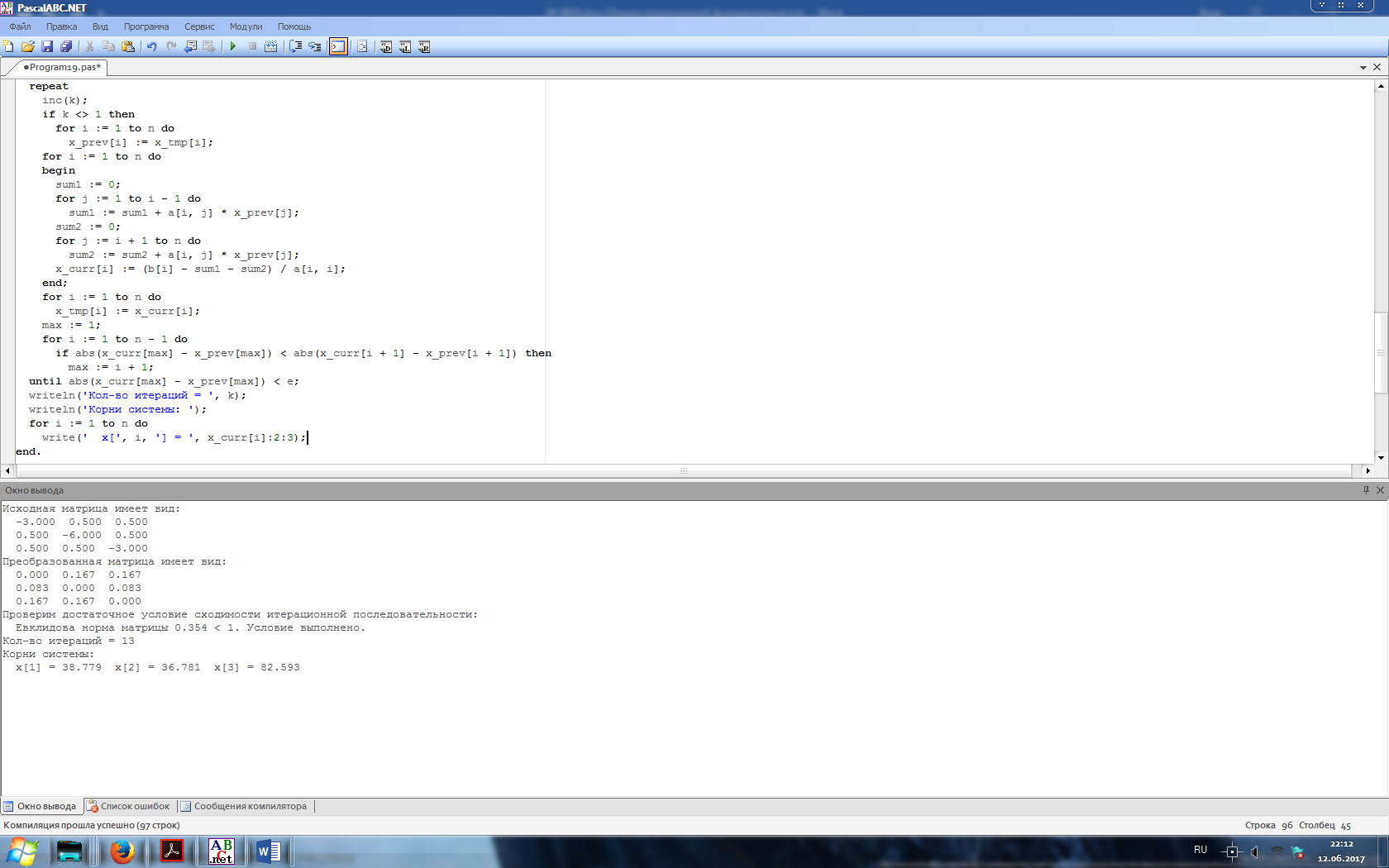
writeln('Кол-во итераций = ', k);

writeln('Корни системы: ');

**for** i := 1 **to** n **do**

write(' x[', i, '] = ', x\_curr[i]:2:3);

**end**.



**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Сформулируйте основные задачи геометрического поиска.   
Поиск в простейшей абстрактной форме, можно представить следующим образом: есть некоторый набор данных (именуемый файлом) и некоторый новый элемент данных (именуемый образцом). Поиск – это установление связи между образцом и файлом. Среди всех форм поиска можно выделить геометрическую форму. Хотя бы, потому что в геометрических приложениях файлы представляют собой сложные структуры, такие как многоугольники, графы и т.п., а в качестве образцов использоваться точки, регионы и т.п.

2. Поясните суть задачи локализации точки и приведите ее алгоритм.   
Одной из главных моделей геометрического поиска является задача локализации точки, когда файл представляет собой разбиение геометрического пространства на области, а запрос является точкой. Локализация состоит в определении области, содержащей запрошенную точку.

3. Поясните суть задачи регионального поиска и приведите ее алгоритм.   
Задача регионального поиска, когда файл содержит набор точек пространства, а запрос есть некая стандартная гео­метрическая фигура, произвольно перемещаемая в этом пространстве (типичный запрос в 3-мерном пространстве это шар или брус). Региональный поиск состоит в извлечении (задачи отчета) или в подсчете числа (задачи подсчета) всех точек внутри запросного региона (области).