МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Частное учреждение образования

«Гродненский колледж бизнеса и права»

**Лабораторная работа № 19**

**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 № 19**

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;

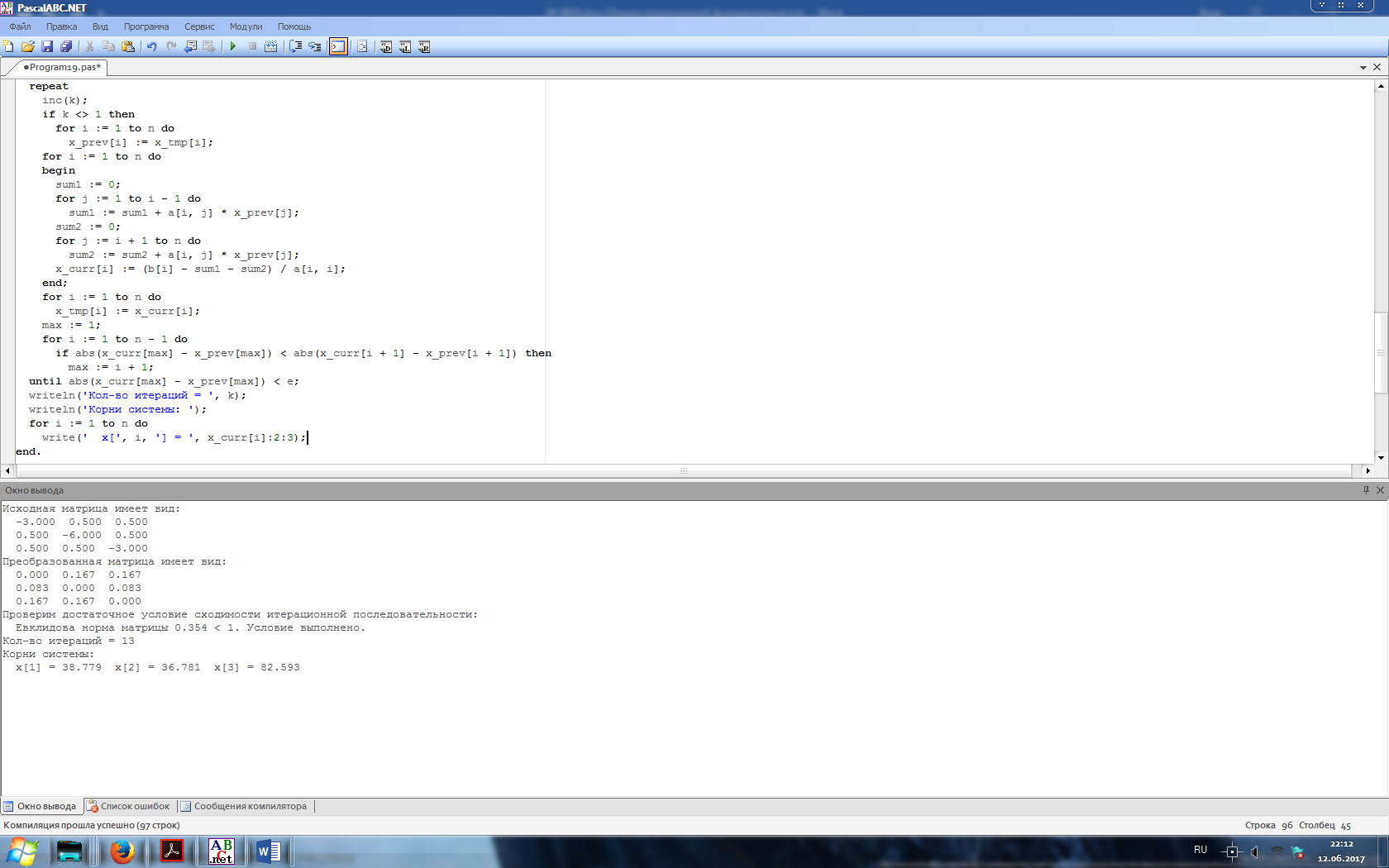
writeln('Кол-во итераций = ', k);

writeln('Корни системы: ');

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

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

**end**.



**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Сформулируй основную задачу вычислительной математики.  
   Решение практических задач, основанных на точности вычислений, различных обусловленностей и зависимостей от алгоритмов или моделей.
2. Приведите алгоритм проверки пересекающихся отрезков.

Шаг 1. Ввод данных (x1;y1) (x2; y2) (x3;y3) (x4; y4)   
Чтобы вычислить правильные угловые коэффициенты, должно выполняться условие x1 ≤ x2; x3 ≤ x4;   
Если нет - то меняем местами пары координат отрезков.   
Шаг 2. Если x1 ≥ x2 то  меняем между собой значения x1 и  x2  и y1 и  y2   
Шаг 3. Если x3 ≥ x4 то  меняем между собой значения x3 и  x4  и y3 и  y4  ;

Шаг 4. Проверяем, равны ли между собой  у2 и у1,   
если у2 = у1 да, то принимаем  k1 = 0 иначе    
Определяем угловой коэффициент в уравнении прямой:   
k1 =  ( у2 - у1 ) / ( x2 - x1 )

Шаг 5. Проверяем, равны ли между собой  у3 и у4,   
если у3 = у4 да, то принимаем  k2 = 0 иначе    
Определяем угловой коэффициент в уравнении прямой:   
k2 =  ( у4 - у3 ) / ( x4 - x3 )

Шаг 6.  Проверим отрезки на параллельность.   
Если k1 = k2 , то прямые параллельны и отрезки пересекаться не могут. Решение задачи прекращаем.

Шаг 7. Вычисляем значения свободных переменных    
Определяем свободные члены в уравнении прямой:     
b1 = у1 - k1 \* x1   
b2 = у3  - k2 \* x3      
Шаг 8. Решаем систему уравнений:     
y = k1 x + b1   
y = k2 x + b2   
  
Если прямые имеют точку пересечения, то    
k1 x + b1 = k2 x + b2   
  
Откуда и вычисляем точку пересечения прямых     
x = ( b2 - b1 ) / ( k1 -  k2 )   
y = k1 x + b1.   
  
Шаг 9. Учтем, что точка пересечения прямых может лежать вне отрезков, принадлежащих этим прямым. Таким образом, если отрезки пересекаются, то, поскольку     
x1 ≤ x2; x3 ≤ x4;   
должны выполняться условия:    
x1  ≤ x4 и x4  ≤ x2   
или   
x1  ≤ x3 и x3  ≤ x2   
Если одно из двух условий верно, то отрезки имеют точку пересечения, иначе - отрезки не пересекаются.