МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Частное учреждение образования

«Гродненский колледж бизнеса и права»

**Лабораторная работа № 21**

**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 № 21**

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;

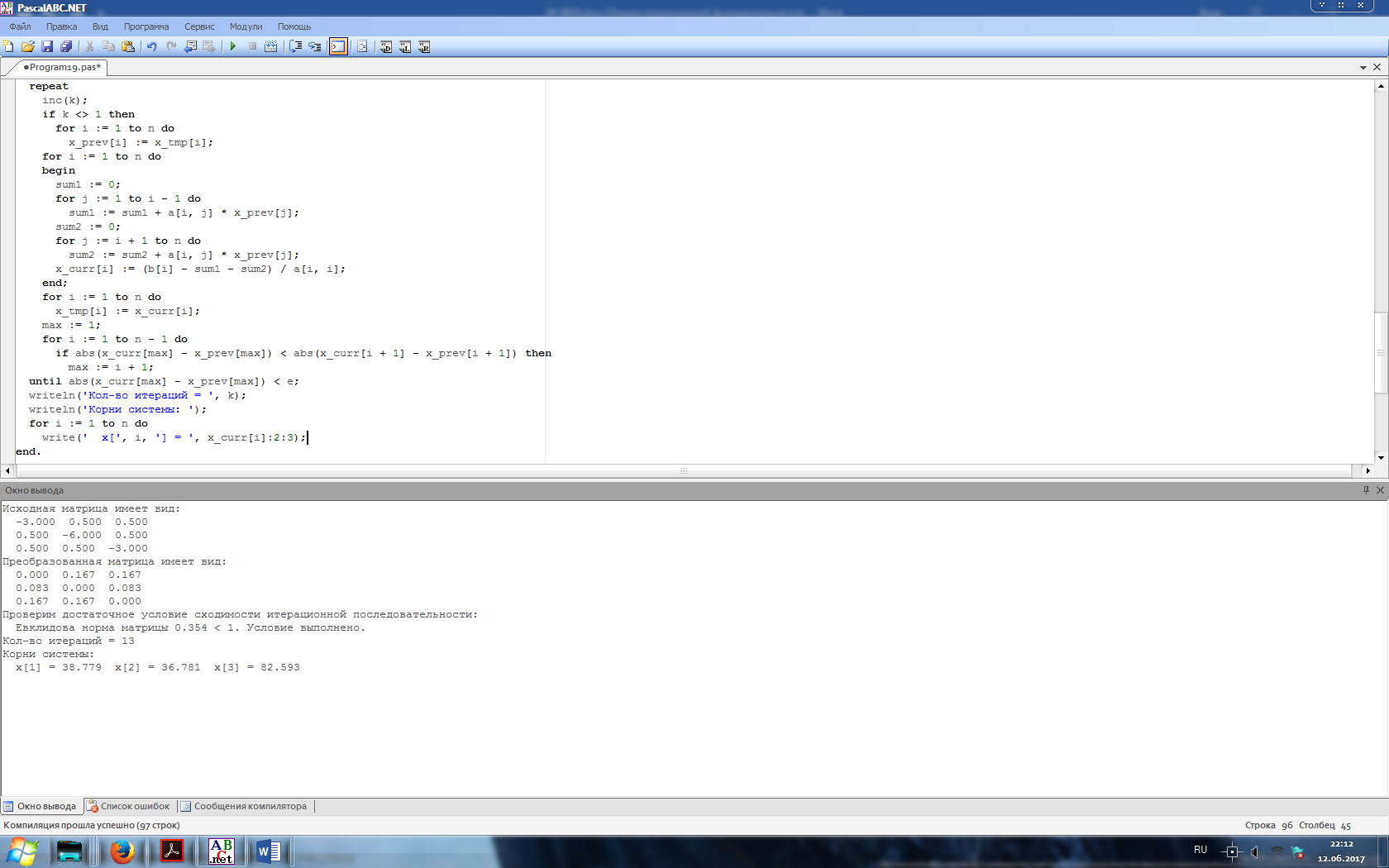
writeln('Кол-во итераций = ', k);

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 методы генерации случайных чисел.   
Все генераторы случайных чисел делятся на два вида:  
‒ True random number generator (ГНСЧ, генератор настоящих случайных чисел);   
‒ Pseudo random number generator (ГПСЧ, генератор псевдослучайных чисел); Эти два генератора задания случайной последовательности отличаются способом получения случайного числа. ГНСЧ использует в качестве механизма получения случайного числа некий физический процесс, который сам по себе является непредсказуемым. Если оцифровать такой процесс (например, шумы атмосферы), то можно использовать его для генератора случайных чисел. ГПСЧ в свою очередь использует математические алгоритмы (полностью производимые компьютером).

2. Поясните суть физических методов генерации случайных чисел.   
Применение физических датчиков позволяет генерировать случайные последовательности, которые не будут коррелированны на сколь угодно длинном расстоянии. Такие последовательности действительно являются случайными, т. к. они не могут быть воспроизведены в заданном порядке, не могут быть повторены в следующем опыте, являются полностью непредсказуемыми.

3. Поясните суть табличных методов генерации случайных чисел.

Табличные ГСЧ в качестве источника случайных чисел используют специальным образом составленные таблицы, содержащие проверенные некоррелированные, то есть никак не зависящие друг от друга, цифры. Достоинство данного метода в том, что он дает действительно случайные числа, так как таблица содержит проверенные некоррелированные цифры. Недостатки метода: для хранения большого количества цифр требуется много памяти; большие трудности порождения и проверки такого рода таблиц, повторы при использовании таблицы уже не гарантируют случайности числовой последовательности, а значит, и надежности результата.

4. Поясните суть алгоритмических методов генерации случайных чисел.   
Числа, генерируемые с помощью этих ГСЧ, всегда являются псевдослучайными (или квазислучайными), то есть каждое последующее сгенерированное число зависит от предыдущего:

*ri*+ 1 = *f*(*ri*)

Последовательности, составленные из таких чисел, образуют петли, то есть обязательно существует цикл, повторяющийся бесконечное число раз. Повторяющиеся циклы называются периодами.

Достоинством данных ГСЧ является быстродействие; генераторы практически не требуют ресурсов памяти, компактны. Недостатки: числа нельзя в полной мере назвать случайными, поскольку между ними имеется зависимость, а также наличие периодов в последовательности квазислучайных чисел.