*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе № 4**

**Дисциплина:** Технологии разработки программных систем

**Название лабораторной работы:**

Оценка эффективности и качества программы

**Вариант 13**

Студент гр. ИУ6-42  **\_\_\_\_\_\_12.05.2018 А.С. Бурлаков**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_ К.С. Хорунжина**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

# ВВЕДЕНИЕ

## 1. Цель работы

* Изучить известные критерии оценки и способы повышения эффективности и качества программных продуктов.

### 2. Порядок выполнения

## Ознакомиться с теоретическими сведениями.

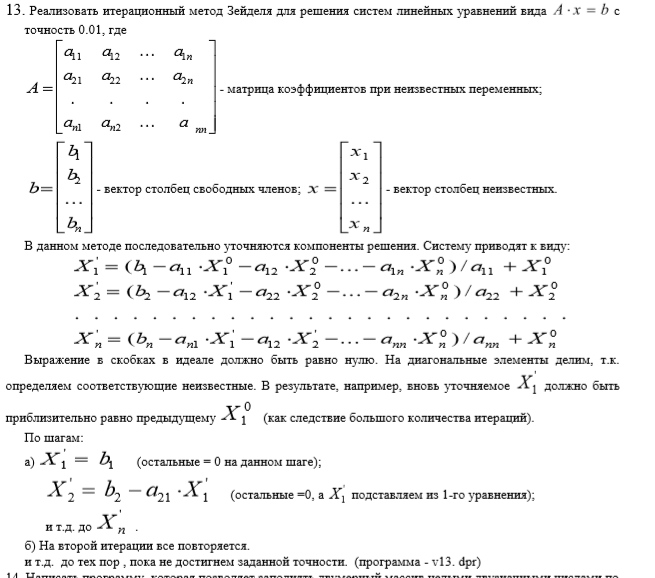
## Исследовать заданный пример программы с целью определения ее эффективности и качества.

## Определить основные критерии оценки и количественные характеристики для заданной программы.

## Предложить варианты повышения эффективности и улучшения качества для заданного примера программы.

## Составить отчет. В отчете должны быть приведены: задача и исходная программа, улучшенный вариант программы и оценочные таблицы.

## 3. Задание



# ОСНОВНАЯ ЧАСТЬ

Листинг исходной программы

program V13;

uses

Timers;

const

Nmax = 10;

var

i, j, k, max, kx, ky, n, y, l, o: integer;

eps: real;

x, b, sw, s: array [1..Nmax] of real;

a: array [1..Nmax, 1..Nmax] of real;

begin

// Ввод n и max

write('Количество уравнений : ');

readln(n);

write('Введите максимальное количество итераций: ');

readln(max);

writeln('Введите числа в матрицу и свободные члены: ');

// Ввод матрицы

eps := 0.01;

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

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

**begin**

write('a', i, j, '=');

readln(a[i, j]);

**end**;

writeln; // Ввод массива b

for i := 1 to n do

begin

write('b', i, '=');

readln(b[i]);

end;

for i := 1 to n do

x[i] := 0;

k := 0;

repeat

k := k + 1;

sw[1] := 0;

for i := 1 to n do

begin

s[i] := b[i];

for j := 1 to n do

s[i] := s[i] - a[i, j] \* x[j];

s[i] := s[i] / a[i, i];

x[i] := s[i] + x[i];

sw[1] := sw[1] + abs(s[i]);

end;

until (k >= max) or (sw[1] < eps);

for i := 1 to n do

write(' x[', i, '] = ', x[i]:4:4);

writeln;

if k < max then

writeln(' k=', k)

else

writeln('Сходимости нет');

end.

## В результате исследования заданной программы сделаны следующие выводы:

1. Программа не оптимальна по объему используемой памяти – для переменных i, j, k используются только неотрицательные значения, переменные kx, ky, y не используются, в массиве sw[10] используется только один элемент – sw[1], массив s[10] не имеет смысла.
2. Программа не содержит защиты от ввода неверного количества уравнений (Если ввести 11 программа завершится с ошибкой)

Листинг улучшенной программы

**program** V13;

**const**

Nmax = 10;

**var**

i, j, k: word;

max, n: integer;

eps, sw, s: real;

x, b: **array** [1..Nmax] **of** real;

a: **array** [1..Nmax, 1..Nmax] **of** real;

**begin**

// Ввод n и max

write('Количество уравнений : ');

readln(n);

**if** (n > Nmax) **then begin**

writeln('Количество уравнений = 10');

n := Nmax;

**end**;

write('Введите максимальное количество итераций: ');

readln(max);

writeln('Введите числа в матрицу и свободные члены: ');

// Ввод матрицы

eps := 0.01;

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

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

**begin**

write('a', i, j, '=');

readln(a[i, j]);

**end**;

writeln;

// Ввод массива b

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

**begin**

write('b', i, '=');

readln(b[i]);

**end**;

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

x[i] := 0;

k := 0;

**repeat**

k := k + 1;

sw := 0;

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

**begin**

s := b[i];

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

s := s - a[i, j] \* x[j];

s := s / a[i, i];

x[i] := s + x[i];

sw := sw + abs(s);

**end**;

**until** (k >= max) **or** (sw < eps);

**end**;

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

write(' x[', i, '] = ', x[i]:4:4);

writeln;

**if** k < max **then**

writeln(' k=', k)

**else**

writeln('Сходимости нет');

writeln(' x=', x);

**end**.

Cписок изменений:

1. У переменных s и sw изменен тип с array of real на real;
2. У переменных i,j,k изменен тип с integer на word;
3. Убраны переменные kx, ky, y;
4. Добавлена проверка ввода количества уравнений.

Таблица 1 – Оценка эффективности программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Исходная программа | | Улучшенная программа | |
| Критерий оценки | Недостатки | Оценка | Улучшения | Оценка |
| Время выполнения | Лишнее обращение к массивам s, sw | 2200ms (миллион запусков) | Изменен тип переменных s, sw | 1500ms (миллион запусков) |
| Оперативная память | У i,j,k не используется половина значений | 862 байт | Изменен тип переменных s, sw, I, j, k | 699 байт |
| Внешняя память | Не используется | - | Не используется | - |

Таблица 2 – Оценка качества программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Исходная программа | | Улучшенная программа | |
| Критерий оценки | Недостатки | Оценка | Улучшения | Оценка |
| Правильность | Код программы полностью соответствует заданию | + | Код программы полностью соответствует заданию | + |
| Универсальность | Программа универсальна | + | Программа универсальна | + |
| Проверяемость | Отсутствует проверка входных данных | - | Входные данные проверяются | + |
| Точность результатов | Вычисления соответствуют заданной точности | + | Вычисления соответствуют заданной точности | + |
| Эффективность | Не эффективно используется память | - | Время выполнения зависит от входных данных | + |

# ЗАКЛЮЧЕНИЕ

В результате данной лабораторной работы изучены основные критери оценки и способы повышения эффективности и качества программ. Предложен вариант повышения эффективности и улучшения качества программы.