**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 Е.С. Мухина

(Подпись)

\_\_\_ \_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** научиться составлять программы для решения численных методов решения обыкновенных дифференциальных уравнений.

**Теоретическая часть**

Дифференциальным уравнением будем называть соотношение типа равенство двух функций, каждая часть которого может содержать независимые переменные, искомую функциональную зависимость и ее производные функции.

Если искомая зависимость, входящая в запись дифференциального уравнения, является функцией одной независимой переменной, то такое уравнение называется обыкновенным дифференциальным уравнением. Если же искомая функция зависит от нескольких независимых переменных, то уравнение называется уравнением в частных производных.

*Численные методы решения систем дифференциальных уравнений.*

1. *Метод Эйлера.*

yij+1=yij+hfi(xi,y1j y2j..ynj)



j - номер шага.

xj+1=xj+h

1. *Модифицированный метод Эйлера.*

ki1=h\*fi(xj,y1j..ynj)

ki1=h\*fi(xj+h,y1j+ki1..ynj+ki2)

yij+1=yij+(ki1+ki2)/2

xj+1=xj+h

1. *Метод Рунге-Кутта четвертого порядка.*

ki1=hfi(xj,y1j..ynj)

ki2=hfi(xj+h/2,y2j+ki1/2,..,ynj+kn1/2)

ki3=hfi(xj+h/2,y2j+ki2/2,..,ynj+kn2/2)

ki4=hfi(xj+h,y1j+ki2,..,ynj+kn3)

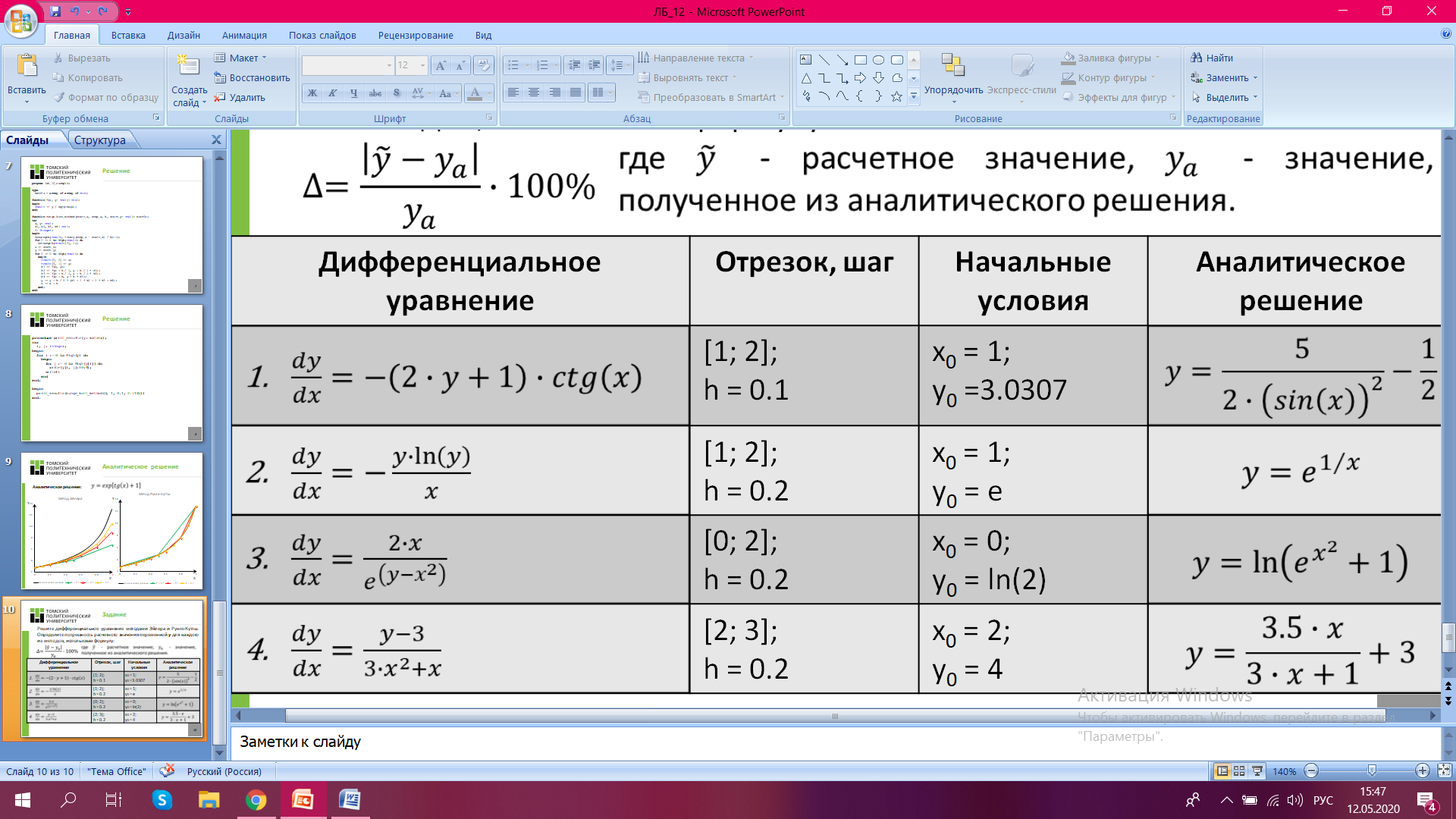
yij+1=yij+(ki1+2ki2+2ki3+ki4)/6

xj+1=xj+h

**Практическая часть**

**Задание**

**Задание:** Решите дифференциальное уравнение методами Эйлера и Рунге-Кутты. Определите погрешность расчетного значения переменной ***y*** для каждого из методов, использовав формулу:

****

**Программная реализация:**

**1)**

**1 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := -(2 \* y + 1) \* 1 / tan(x)

**end**;

**function** fa(x: real): real;

**begin**

result := 5 / (2 \* sqr(sin(x))) - 1 / 2

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(eyler\_method(1, 2, 0.1, 3.0307)))

**end**.

**Ответ:**

  1.0000    3.0307    0.0002

    1.1000    2.5773    2.6563

    1.2000    2.2640    4.7870

    1.3000    2.0491    6.5469

    1.4000    1.9076    8.0402

    1.5000    1.8245    9.3430

    1.6000    1.7916   10.5170

    1.7000    1.8050   11.6172

    1.8000    1.8649   12.6973

    1.9000    1.9752   13.8141

    2.0000    2.1443   15.0302

**2 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := -(2 \* y + 1) \* 1 / tan(x)

**end**;

**function** fa(x: real): real;

**begin**

result := 5 / (2 \* sqr(sin(x))) - 1 / 2

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h)+1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(runge\_kutt\_method(1, 2, 0.1, 3.0307)))

**end**.

**Ответ:**

1.0000 3.0307 0.0002

1.1000 2.6476 0.0003

1.2000 2.3779 0.0003

1.3000 2.1927 0.0003

1.4000 2.0744 0.0003

1.5000 2.0126 0.0003

1.6000 2.0021 0.0003

1.7000 2.0422 0.0003

1.8000 2.1361 0.0003

1.9000 2.2918 0.0003

2.0000 2.5236 0.0003

**2)**

**1 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := - y \* ln(y) / x

**end**;

**function**fa(x: real): real;

**begin**

result := exp(1 / x)

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(eyler\_method(1, 2, 0.2, exp(1))))

**end**.

**Ответ:**

   1.0000    2.7183    0.0000

    1.2000    2.1746    5.4912

    1.4000    1.8931    7.3267

    1.6000    1.7205    7.9098

    1.8000    1.6038    7.9825

    2.0000    1.5196    7.8312

**2 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := - y \* ln(y) / x

**end**;

**function** fa(x: real): real;

**begin**

result := exp(1 / x)

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h)+1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(runge\_kutt\_method(1, 2, 0.2, exp(1))))

**end**.

**Ответ:**

1.0000 2.7183 0.0000

1.2000 2.3010 0.0003

1.4000 2.0427 0.0004

1.6000 1.8683 0.0004

1.8000 1.7429 0.0003

2.0000 1.6487 0.0003

**3)**

**1 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := 2 \* x / exp(y - sqr(x))

**end**;

**function**fa(x: real): real;

**begin**

result := ln(exp(sqr(x)) + 1)

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(eyler\_method(0, 2, 0.2, ln(2))))

**end**.

**Ответ:**

0.0000    0.6931    0.0000

    0.2000    0.6931    2.8317

    0.4000    0.7348    5.3538

    0.6000    0.8248    7.2451

    0.8000    0.9756    8.2639

    1.0000    1.2044    8.2910

    1.2000    1.5304    7.3937

    1.4000    1.9689    5.8728

    1.6000    2.5240    4.1947

    1.8000    3.1874    2.7749

    2.0000    3.9463    1.7882

**2 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := 2 \* x / exp(y - sqr(x))

**end**;

**function** fa(x: real): real;

**begin**

result := ln(exp(sqr(x)) + 1)

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h)+1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(runge\_kutt\_method(0, 2, 0.2, ln(2))))

**end**.

**Ответ:**

0.0000 0.6931 0.0000

0.2000 0.7133 0.0001

0.4000 0.7763 0.0006

0.6000 0.8893 0.0015

0.8000 1.0635 0.0031

1.0000 1.3133 0.0055

1.2000 1.6528 0.0089

1.4000 2.0920 0.0128

1.6000 2.6349 0.0164

1.8000 3.2790 0.0190

2.0000 4.0190 0.0204

**4)**

**1 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := (y - 3) / (3 \* sqr(x) + x)

**end**;

**function** fa(x: real): real;

**begin**

result := (3.5 \* x) / (3 \* x + 1) + 3

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(eyler\_method(2, 3, 0.2, 4)))

**end**.

**Ответ:**

2.0000 4.0000 0.0000

2.2000 4.0143 0.0281

2.4000 4.0264 0.0504

2.6000 4.0368 0.0684

2.8000 4.0459 0.0831

3.0000 4.0539 0.0953

**2 способ:**

**Program** lb12;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result :=(y - 3) / (3 \* sqr(x) + x)

**end**;

**function** fa(x: real): real;

**begin**

result := (3.5 \* x) / (3 \* x + 1) + 3

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h)+1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** get\_error(y: matrix): matrix;

**var**

i, j: integer;

**begin**

SetLength(result, Length(y));

**for** i := 0 **to** High(y) **do**

SetLength(result[i], 3);

**for** i := 0 **to** High(y) **do**

**begin**

result[i, 2] := abs(y[i, 1] - fa(y[i, 0])) / fa(y[i, 0]) \* 100;

**for** j := 0 **to** 1 **do**

result[i, j] := y[i, j];

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:10:4);

writeln

**end**

**end**;

**begin**

print\_results(get\_error(runge\_kutt\_method(2, 3, 0.2, 4)))

**end**.

**Ответ:**

2.0000 4.0000 0.0000

2.2000 4.0132 0.0000

2.4000 4.0244 0.0000

2.6000 4.0341 0.0000

2.8000 4.0426 0.0000

3.0000 4.0500 0.0000

**Выводы**

В ходе работы могу смело утверждать, что я научилась составлять программы для решения численных методов решения обыкновенных дифференциальных уравнений.