Условие:

```
11-2 ( 1 10) Решить задачу Коши дла ОДУ порядка 2:
          y''= -  2* y'  -  3 * y  +  3 * x  +  3
        -1 <= x <=   1  ; y(-1 ) =  2.0 ;  y'(-1 ) = -7.0
     *)методом Эйлера  с шагами h1= 0.500 ,  h2= 0.250 и  h3= 0.200
        уточнить y(  1 )   по формуле Рунге
     *)методом Рунге-Кутты(4) с шагами h1= 0.500 и  h2= 0.250
        уточнить y(  1 )   по формуле Рунге-Ромберга
```

Теория:

12.2. Решение задачи Коши для ОДУ второго порядка

Рассмотрим задачу Коши для дифференциального уравнения второго порядка, разрешённого относительно второй производной:

$$y'' = f(x,y,y')\,,\ x\in[a;b]\ ,\ y(a) = c\,,\ y'(a) = d\ . \qquad (12.2.1)$$

С помощью замены $z=y'$ уравнение (11.2-1) превратится в систему

$$\begin{cases} y' = z \\ z' = f(x,y,z) \end{cases},\ x\in[a;b]\ ,\ y(a) = c\,,\ z(a) = d\ . \qquad (12.2.2)$$

Аналогично дифференциальное уравнение n- го порядка сводится к системе n- го порядка (n уравнений, каждое уравнение первого порядка).

Численное решение (12.2.2) заключается в построении (вычислении) таблицы x-y-z , в которой x меняется от a до b с некоторым шагом (одинаковым или разным). Рассмотрим те же два метода решения этой задачи: метод Эйлера (первого порядка) и метод Рунге–Кутты (четвёртого порядка).

12.2.1. Метод Эйлера для ОДУ второго порядка

Если известно y_i и z_i = значения табличной функции x-y-z при $x=x_i$, то можно вычислить новый узел $x=x_{i+1}$ и ему соответствующие y_{i+1} и z_{i+1} по формуле Эйлера для системы (11.2.2):

$$\begin{aligned} x_{i+1} &= x_i + h \\ y_{i+1} &= y_i + h\cdot f_1(x_i;y_i,z_i) = y_i + h\cdot z_i\ , \\ z_{i+1} &= z_i + h\cdot f_2(x_i;y_i,z_i) = z_i + h\cdot f(x_i;y_i,z_i)\ . \end{aligned} \qquad (12.2.1.1)$$

Для последнего $x_n=b$ вычисляется только y_n и таблица x-y (а если необходимо и z) готова.

В формулах (12.2.1.1) шаг постоянен, но на практике шаг можно менять.

12.2.2. Метод Рунге–Кутты для ОДУ второго порядка

Если известно y_i и z_i = значения табличной функции x-y-z при $x=x_i$, то для системы (11.2-2) можно вычислить новый узел $x=x_{i+1} = x_i+h$ и ему соответствующие y_{i+1} и z_{i+1} по формуле Рунге-Кутты (четвёртого/ порядка), вычислив сначала восемь промежуточных коэффициентов (четыре пары):

$$K_{1y}= z_i \ , \qquad K_{1z}=f\,(\,x_i;\ y_i;\ z_i),$$
$$K_{2y}= z_i+h/2\cdot K_{1z} \ , \qquad K_{2z}=f\,(\,x_i+h/2\ ;\ y_i+h/2\cdot K_{1y};\ z_i+h/2\cdot K_{1z}\,)\ ,$$
$$K_{3y}= z_i+h/2\cdot K_{2z} \ , \qquad K_{3z}=f\,(\,x_i+h/2\ ;\ y_i+h/2\cdot K_{2y}\ ;\ z_i+h/2\cdot K_{2z}\,)\ ,$$
$$K_{4y}= z_i+h\cdot K_{3z} \ , \qquad K_{4z}=f\,(\,x_i+h\ ;\ y_i+h\cdot K_{3y}\ ;\ z_i+h\cdot K_{3z}\,)\ .$$

После этого вычисляется y_{i+1} и z_{i+1} в новой точке $x_{i+1} = x_i + h$ по формулам :

$$y_{i+1} = y_i + h/6\cdot(\,K_{1y}+2\cdot K_{2y}+2\cdot K_{3y}+K_{4y}\,)\ ,$$
$$z_{i+1} = z_i + h/6\cdot(\,K_{1z}+2\cdot K_{2z}+2\cdot K_{3z}+K_{4z}\,)\ .$$

Для последнего $x_n=b$ вычисляется только y_n и таблица x-y (а если необходимо и z) готова.

Решение:

EulerMethodSecond.java

```
package fourth;

abstract class EulerMethodSecond {
    abstract double f(double x, double y, double z);
    abstract double a();
    abstract double b();
    abstract double ya();
    abstract double yda();
    abstract double h();
    void exec(){
        int ic = (int)((b() - a())/h());
        ++ic;
        double[] x = new double[ic];
        double[] y = new double[ic];
        double[] z = new double[ic];

        x[0] = a();
        y[0] = ya();
        z[0] = yda();

        for(int i = 1;i<ic;++i){
            x[i] = x[i-1] + h();
            y[i] = y[i-1] + h()*z[i-1];
            z[i] = z[i-1] + h()*f(x[i-1],y[i-1], z[i-1]);
        }
        printRes(ic, x,y,z);
    }
    void printRes(int ic, double[] x, double[]y, double[] z){
        System.out.println("Метод Эйлера для решения ОДУ второго порядка");
        System.out.println("Шаг: "+h());
        for(int i =0 ;i<ic;++i){
            System.out.printf("i: %d, x: %.9f, y: %.9f, z: %.9f\n", i,
x[i],y[i],z[i]);
        }
```

```
    }
}
```

Main.java

```
package fourth;

public class Main {
    public static void main(String[] args){
        EulerMethodSecond task1 = new EulerMethodSecond() {
            @Override
            double f(double x, double y, double z) { return -2*z - 3 * y +
3*x + 3;}
            @Override
            double a() { return -1;}
            @Override
            double b() { return 1;}
            @Override
            double ya() { return 2;}
            @Override
            double yda() { return -7;}
            @Override
            double h() { return 0.5;}
        };
        task1.exec();

        EulerMethodSecond task2 = new EulerMethodSecond() {
            @Override
            double f(double x, double y, double z) {return -2*z - 3 * y +
3*x + 3; }
            @Override
            double a() { return -1;}
            @Override
            double b() { return 1;}
            @Override
            double ya() { return 2; }
            @Override
            double yda() { return -7;}
            @Override
            double h() { return 0.25;}
        };
        task2.exec();

        EulerMethodSecond task3 = new EulerMethodSecond() {
            @Override
            double f(double x, double y, double z) {return -2*z - 3 * y +
3*x + 3; }
            @Override
            double a() {return -1;}
            @Override
            double b() {return 1; }
            @Override
            double ya() { return 2;}
            @Override
            double yda() {return -7;}
            @Override
            double h() {return 0.2;}
        };
        task3.exec();

        RungeKuttaMethodSecond task4 = new RungeKuttaMethodSecond() {
            @Override
```

```
            double f(double x, double y, double z) {return -2*z - 3 * y +
3*x + 3; }
            @Override
            double a() {return -1;}
            @Override
            double b() {return 1; }
            @Override
            double ya() { return 2;}
            @Override
            double yda() {return -7;}
            @Override
            double h() {return 0.5;}
        };
        task4.exec();

        RungeKuttaMethodSecond task5 = new RungeKuttaMethodSecond() {
            @Override
            double f(double x, double y, double z) {return -2*z - 3 * y +
3*x + 3; }
            @Override
            double a() {return -1;}
            @Override
            double b() {return 1; }
            @Override
            double ya() { return 2;}
            @Override
            double yda() {return -7;}
            @Override
            double h() {return 0.25;}
        };
        task5.exec();
    }
}
```

RungeKuttaMethodSecond.java

```
package fourth;

abstract public class RungeKuttaMethodSecond {
    abstract double a();
    abstract double b();
    abstract double h();
    abstract double f(double x, double y, double z);
    abstract double ya();
    abstract double yda();
    void exec(){
        int ic = (int)((b()-a())/h())+1;
        double x[] = new double[ic];
        double y[] = new double[ic];
        double z[] = new double[ic];
        x[0] = a();
        y[0] = ya();
        z[0] = yda();
        System.out.println("Метод Рунге-Кутта для ОДУ второго порядка");
        System.out.println("Шаг: " + h());
        for(int i = 1;i<ic;++i){
           double X = x[i-1];
           double Y = y[i-1];
           double Z = z[i-1];
           double K1y = Z;
           double K1z = f(X,Y,Z);
           double K2y = Z + (h()/2)*K1z;
           double K2z = f(X + h()/2, Y+h()*K1y/2, Z + h()*K1z/2);
           double K3y = Z + h()*K2z/2;
           double K3z = f(X + h()/2, Y + h()*K2y/2, Z + h()*K2z/2);
```

```
            double K4y = Z + h() * K3z;
            double K4z = f(X + h(), Y + h() * K3y, Z + h() * K3z);

             y[i] = Y + h() / 6 * (K1y + 2*K2y + 2*K3y + K4y);
             z[i] = Z + h() / 6 * (K1z + 2*K2z + 2*K3z + K4z);
             x[i] = X + h();
             printRes(i, x[i],y[i],z[i], K1y,K1z,K2y,K2z,K3y,K3z,K4y,K4z);
        }
    }
    void printRes(int i, double x,
                  double y, double z,
                  double K1y, double K1z,
                  double K2y, double K2z,
                  double K3y, double K3z,
                  double K4y, double K4z){
        System.out.printf("i: %d, x: %.9f, y: %.9f, z: %.9f, K1y: %.9f, K1z:
%.9f, K2y: %.9f, K2z: %.9f, K3y: %.9f, K3z: %.9f, K4y: %.9f, K4z: %.9f\n",
                i,x,y,z, K1y,K1z,K2y,K2z,K3y,K3z,K4y,K4z);
    }
}
```

Вывод консоли:

```
Метод Эйлера для решения ОДУ второго порядка
Шаг: 0.5
i: 0, x: -1,000000000, y: 2,000000000, z: -7,000000000
i: 1, x: -0,500000000, y: -1,500000000, z: -3,000000000
i: 2, x: 0,000000000, y: -3,000000000, z: 3,000000000
i: 3, x: 0,500000000, y: -1,500000000, z: 6,000000000
i: 4, x: 1,000000000, y: 1,500000000, z: 4,500000000
Метод Эйлера для решения ОДУ второго порядка
Шаг: 0.25
i: 0, x: -1,000000000, y: 2,000000000, z: -7,000000000
i: 1, x: -0,750000000, y: 0,250000000, z: -5,000000000
i: 2, x: -0,500000000, y: -1,000000000, z: -2,500000000
i: 3, x: -0,250000000, y: -1,625000000, z: -0,125000000
i: 4, x: 0,000000000, y: -1,656250000, z: 1,718750000
i: 5, x: 0,250000000, y: -1,226562500, z: 2,851562500
i: 6, x: 0,500000000, y: -0,513671875, z: 3,283203125
i: 7, x: 0,750000000, y: 0,307128906, z: 3,151855469
i: 8, x: 1,000000000, y: 1,095092773, z: 2,658081055
Метод Эйлера для решения ОДУ второго порядка
Шаг: 0.2
i: 0, x: -1,000000000, y: 2,000000000, z: -7,000000000
i: 1, x: -0,800000000, y: 0,600000000, z: -5,400000000
i: 2, x: -0,600000000, y: -0,480000000, z: -3,480000000
i: 3, x: -0,400000000, y: -1,176000000, z: -1,560000000
i: 4, x: -0,200000000, y: -1,488000000, z: 0,129600000
i: 5, x: -0,000000000, y: -1,462080000, z: 1,450560000
i: 6, x: 0,200000000, y: -1,171968000, z: 2,347584000
i: 7, x: 0,400000000, y: -0,702451200, z: 2,831731200
i: 8, x: 0,600000000, y: -0,136104960, z: 2,960509440
i: 9, x: 0,800000000, y: 0,455996928, z: 2,817968640
i: 10, x: 1,000000000, y: 1,019590656, z: 2,497183027
```

Метод Рунге-Кутта для ОДУ второго порядка
Шаг: 0.5
i: 1, x: -0,500000000, y: -0,437500000, z: -2,687500000, K1y: -7,000000000, K1z: 8,000000000, K2y: -5,000000000, K2z: 10,000000000, K3y: -4,500000000, K3z: 7,500000000, K4y: -3,250000000, K4z: 8,750000000
i: 2, x: 0,000000000, y: -0,904785156, z: 0,565917969, K1y: -2,687500000, K1z: 8,187500000, K2y: -0,640625000, K2z: 6,859375000, K3y: -0,972656250, K3z: 5,988281250, K4y: 0,306640625, K4z: 5,158203125
i: 3, x: 0,500000000, y: -0,207668304, z: 1,966663361, K1y: 0,565917969, K1z: 4,582519531, K2y: 1,711547852, K2z: 2,616821289, K3y: 1,220123291, K3z: 2,740447998, K4y: 1,936141968, K4z: 1,511886597
i: 4, x: 1,000000000, y: 0,832589000, z: 2,052042454, K1y: 1,966663361, K1z: 1,189678192, K2y: 2,264082909, K2z: -0,130158424, K3y: 1,934123755, K3z: 0,306695223, K4y: 2,120010972, K4z: -0,518202662

Метод Рунге-Кутта для ОДУ второго порядка
Шаг: 0.25
i: 1, x: -0,750000000, y: 0,514322917, z: -4,845052083, K1y: -7,000000000, K1z: 8,000000000, K2y: -6,000000000, K2z: 9,000000000, K3y: -5,875000000, K3z: 8,375000000, K4y: -4,906250000, K4z: 8,968750000
i: 2, x: -0,500000000, y: -0,423839357, z: -2,688378440, K1y: -4,845052083, K1z: 8,897135417, K2y: -3,732910156, K2z: 8,864746094, K3y: -3,736958822, K3z: 8,455790202, K4y: -2,731104533, K4z: 8,221959432
i: 3, x: -0,250000000, y: -0,857199566, z: -0,844097022, K1y: -2,688378440, K1z: 8,148274952, K2y: -1,669844071, K2z: 7,494348129, K3y: -1,751584924, K3z: 7,275879446, K4y: -0,869408579, K4z: 6,574023921
i: 4, x: 0,000000000, y: -0,885032300, z: 0,542650366, K1y: -0,844097022, K1z: 6,509792742, K2y: -0,030372930, K2z: 5,573880940, K3y: -0,147361905, K3z: 5,502712356, K4y: 0,531581067, K4z: 4,618957993
i: 5, x: 0,250000000, y: -0,626494405, z: 1,450316542, K1y: 0,542650366, K1z: 4,569796168, K2y: 1,113874887, K2z: 3,598853239, K3y: 0,992507021, K3z: 3,627379776, K4y: 1,449495310, K4z: 2,761726014
i: 6, x: 0,500000000, y: -0,195485651, z: 1,935458056, K1y: 1,450316542, K1z: 2,728850131, K2y: 1,791422808, K2z: 1,877768895, K3y: 1,685037654, K3z: 1,962624354, K4y: 1,940972630, K4z: 1,233759713
i: 7, x: 0,750000000, y: 0,313836682, z: 2,093789375, K1y: 1,935458056, K1z: 1,215540840, K2y: 2,087400661, K2z: 0,560858859, K3y: 2,005565413, K3z: 0,667550878, K4y: 2,102345775, K4z: 0,127591341
i: 8, x: 1,000000000, y: 0,832975359, z: 2,030867445, K1y: 2,093789375, K1z: 0,120911205, K2y: 2,108903276, K2z: -

```
0,319487612, K3y: 2,053853423, K3z: -0,215055621, K4y:
2,040025470, K4z: -0,561951053
Process finished with exit code 0
```

Уточнение решений

Выполним уточнение по формуле Рунге.

$h_1 = 0.5 \quad h_2 = 0.25 \quad h_3 = 0.2$

$z_1 = 1.5 \quad z_2 = 1.095092773 \quad z_3 = 1019590656 \quad p = 1$

$$D_1 = \begin{pmatrix} 1.5 & 0.5 & 0.25 \\ 1.095092773 & 0.25 & 0.0625 \\ 1.019590656 & 0.2 & 0.04 \end{pmatrix}$$

$|D_1| = -0.00275942481$

$$D_2 = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 1 & 0.25 & 0.0625 \\ 1 & 0.2 & 0.04 \end{pmatrix} \qquad |D_2| = -0.00375$$

$$z_p = \frac{|D_1|}{|D_2|} = \frac{-0.00275942481}{-0.00375} = 0{,}735846616$$

Выполним уточнение по формуле Рунге-Ромберга.

$h_1 = 0.5 \quad h_2 = 0.25 \quad y_1 = 0.832589 \quad y_2 = 0{,}832975359 \quad p = 4$

$$z_{pp} = y_1 + \frac{y_1 - y_2}{R^p - 1} \qquad R = \frac{h_2}{h_1} = \frac{0.25}{0.5} = 0.5 \qquad R^p = 0.5^4 = 0.0625$$

$$z_{pp} = 0.832589 + \frac{0.832589 - 0{,}832975359}{0.0625 \; - 1} = 0.8330111667$$

Ответ:

Метод Эйлера

Шаг: 0.5

i	x	y	z
0	-1.0	2.0	-7.0
1	-0.5	-1.5	-3.0
2	0.0	-3.0	3.0
3	0.5	-1.5	6.0
4	1.0	1.5	4.5

Шаг: 0.25

i	x	y	z
0	-1,0	2,0	-7,0
1	-0,75	0,25	-5,0
2	-0,5	-1,0	-2,5
3	-0,25	-1,625	-0,125
4	0,0	-1,65625	1,71875
5	0,25	-1,2265625	2,8515625
6	0,5	-0,513671875	3,283203125
7	0,75	0,307128906	3,151855469
8	1,0	1,095092773	2,658081055

Шаг: 0,2

i	x	y	z
0	-1,0	2,0	-7,0
1	-0,8	0,6	-5,4
2	-0,6	-0,48	-3,48
3	-0,4	-1,176	-1,56
4	-0,2	-1,488	0,1296
5	0,0	-1,46208	1,45056
6	0,2	-1,171968	2,347584
7	0,4	-0,7024512	2,8317312
8	0,6	-0,13610496	2,96050944
9	0,8	0,455996928	2,81796864
10	1,0	1,019590656	2,497183027

Значение y(1) после уточнения по формуле Рунге : 0.735846616

Метод Рунге-Кутты

Шаг: 0.5

i	1	2	3	4
K_{1y}	-7,0	-2,6875	0,565917969	1,966663361
K_{1z}	8,0	8,1875	4,582519531	1,189678192
K_{2y}	-5,0	-0,640625	1,711547852	2,264082909
K_{2z}	10,0	6,859375	2,616821289	-0,130158424
K_{3y}	-4,5	-0,972656250	1,220123291	1,934123755
K_{3z}	7,5	5,988281250	2,740447998	0,306695223
K_{4y}	-3,25	0,306640625	1,936141968	2,120010972
K_{4z}	8,75	5,158203125	1,511886597	-0,518202662
X	-0,5	0,0	0,5	1,0
Y	-0,4375	-0,904785156	-0,207668304	0,832589000
Z	-2,6875	0,565917969	1,966663361	2,052042454

Шаг: 0.25

i	1	2	3	4
K_{1y}	-7,0	-4,845052083	-2,68837844	-0,844097022
K_{1z}	8,0	8,897135417	8,148274952	6,509792742
K_{2y}	-6,0	-3,732910156	-1,669844071	-0,030372930
K_{2z}	9,0	8,864746094	7,494348129	5,573880940
K_{3y}	-5,875	-3,736958822	-1,751584924	-0,147361905
K_{3z}	8,375	8,455790202	7,275879446	5,502712356
K_{4y}	-4,90625	-2,731104533	-0,869408579	0,531581067
K_{4z}	8,968750	8,221959432	6,574023921	4,618957993
X	-0,75	-0,5	-0,25	0,0
Y	0,514322917	-0,423839357	-0,857199566	-0,8850323
Z	-4,845052083	-2,688378440	-0,844097022	0,542650366

i	5	6	7	8
K_{1y}	0,542650366	1,450316542	1,935458056	2,093789375
K_{1z}	4,569796168	2,728850131	1,215540840	0,120911205
K_{2y}	1,113874887	1,791422808	2,087400661	2,108903276
K_{2z}	3,598853239	1,877768895	0,560858859	-0,319487612
K_{3y}	0,992507021	1,685037654	2,005565413	2,053853423
K_{3z}	3,627379776	1,962624354	0,667550878	-0,215055621
K_{4y}	1,449495310	1,940972630	2,102345775	2,040025470
K_{4z}	2,761726014	1,233759713	0,127591341	-0,561951053
X	0,25	0,5	0,75	1,0
Y	-0,626494405	-0,195485651	0,313836682	0,832975359
Z	1,450316542	1,935458056	2,093789375	2,030867445

Значение y(1) после уточнения по формуле Рунге-Бомберга: 0.8330111667