

### Практическое занятие №3

#### Модель Харрода - Домара. Вывод модели Солоу - Свана.

##### Вариант 8

Формулировка задания: Написать программу для расчета объем выпуска, используя соотношения дискретной модели Харрода-Домара и значения констант. Построить графики функций  $Y(t)$ ,  $C(t)$  и  $I(t)$  на одном графике.

Значения констант:

Норма сбережения, S	Средняя производительность капитала, $\sigma$	Начальное значение выпуска, $Y_0$	Расчетный интервал, T
0,58	1,43	100	11

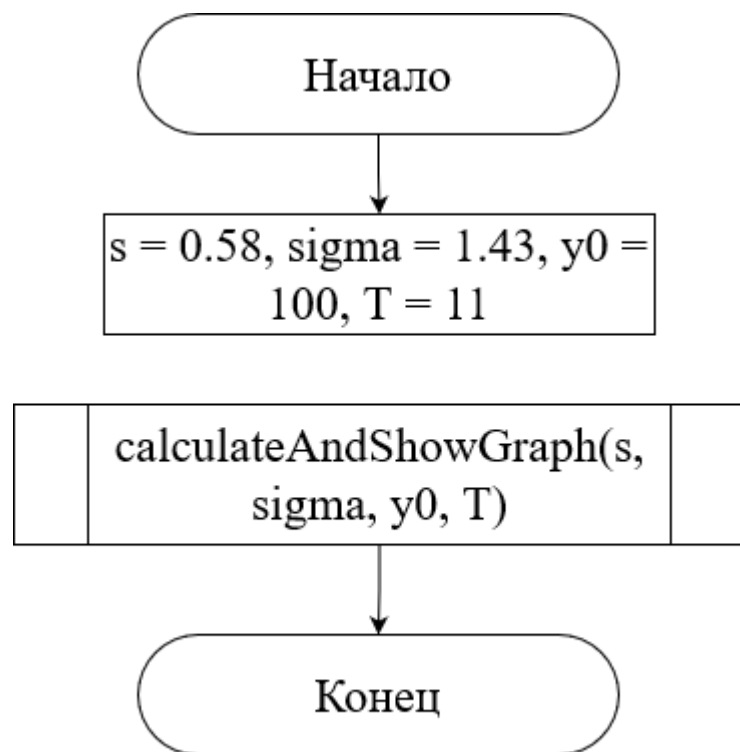
Значения констант должны передаваться в функцию в виде аргументов, эндогенные значения инициализируются нулями.

Словесное описание алгоритма:

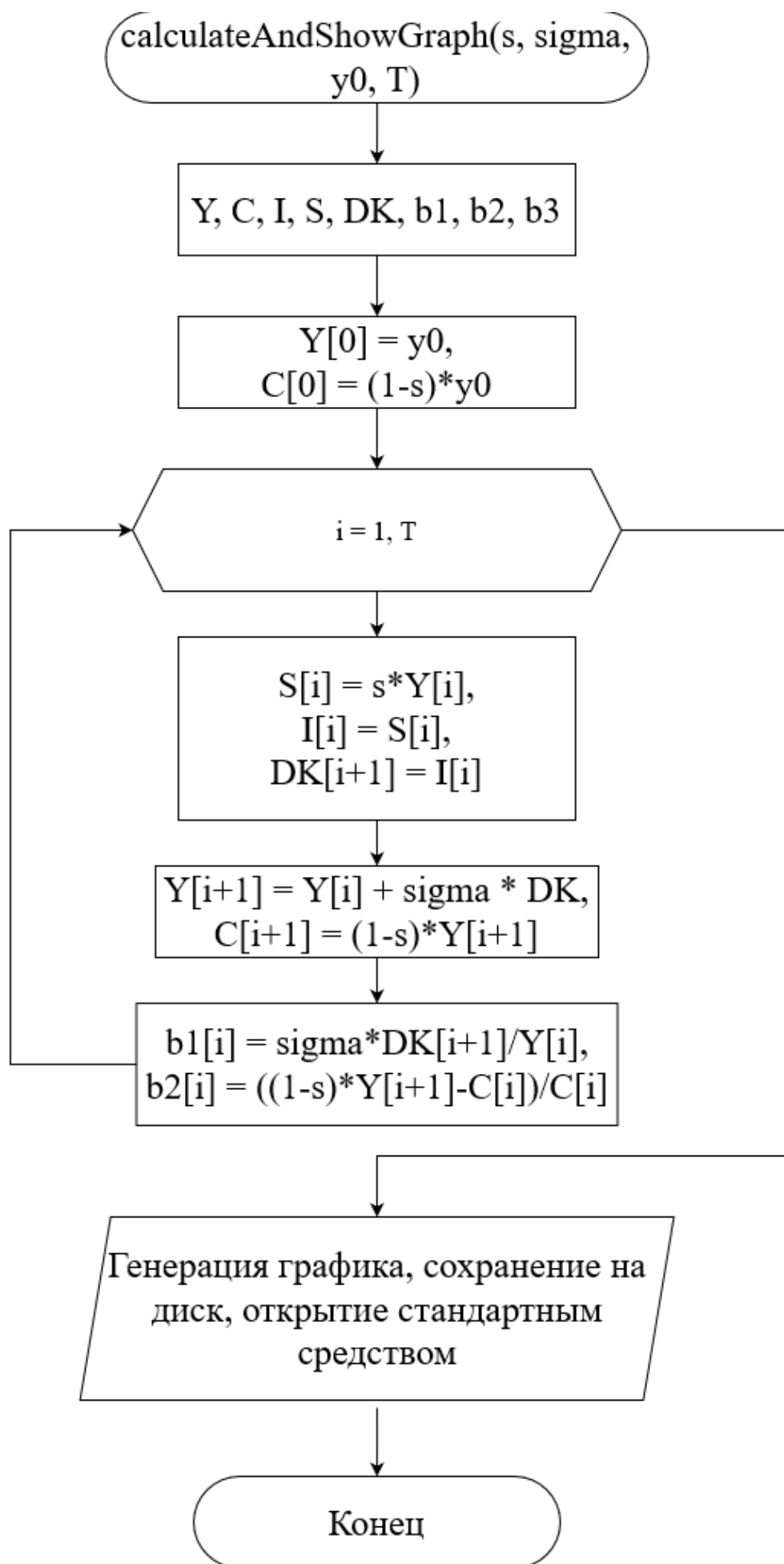
- 1) Получить из входных параметров значения констант
- 2) Инициализировать эндогенные переменные нулями
- 3) Присвоить начальные значения  $Y[0]$  и  $C[0]$
- 4) Открыть цикл от 1 до предела расчетного интервала
- 5) Рассчитать очередные значения S, I и DK
- 6) Рассчитать следующие значения I и C
- 7) Рассчитать значения темпов прироста предложения, потребления и инвестиций
- 8) Закончить цикл
- 9) Сгенерировать график и вывести его

Блок схема функций

Основная функция



Функция расчета и показа графика



## Код программы

```
public class Main {  
  
    public static void main(String[] args) {  
        Modeller m = new Modeller();  
        m.execute();  
    }  
}
```

```
import org.jfree.chart.ChartFactory;  
import org.jfree.chart.ChartUtilities;  
import org.jfree.chart.JFreeChart;  
import org.jfree.data.Range;  
import org.jfree.data.xy.XYSeries;  
import org.jfree.data.xy.XYSeriesCollection;  
  
import java.awt.*;  
import java.io.File;  
import java.io.IOException;  
import java.util.*;  
import java.util.List;  
  
public class Modeller {  
  
    public Modeller() {  
    }  
  
    public void execute() {  
        double s = 0.58, sigma = 1.43, y0 = 100;  
        int T = 11;  
        calculateAndShowGraph(s, sigma, y0, T);  
    }  
  
    private void calculateAndShowGraph(double s, double sigma, double y0, int T) {  
        List<Double> Y = initWithZeroes(T);  
        List<Double> C = initWithZeroes(T);  
        List<Double> I = initWithZeroes(T);  
  
        List<Double> S = initWithZeroes(T);  
        List<Double> DK = initWithZeroes(T);  
  
        List<Double> b1 = initWithZeroes(T);  
        List<Double> b2 = initWithZeroes(T);  
        List<Double> b3 = initWithZeroes(T);  
  
        Y.set(0, y0);  
        C.set(0, (1-s)*y0);  
        for(int i = 0; i <= T; i++) {  
            S.set(i, s*Y.get(i));  
            I.set(i, S.get(i));  
            DK.set(i+1, I.get(i));  
        }  
    }  
}
```

```

        Y.set(i+1, Y.get(i) + sigma*DK.get(i+1));
        C.set(i+1, (1-s)*Y.get(i+1));

        b1.set(i, sigma * DK.get(i+1)/Y.get(i));
        b2.set(i, ((1-s)*Y.get(i+1)-C.get(i))/C.get(i));
        b3.set(i, (s*Y.get(i+1)-s*Y.get(i))/I.get(i));
    }
    final XYSeriesCollection dataset = new XYSeriesCollection();
    dataset.addSeries(listToSeries(Y, "Y(t)", T));
    dataset.addSeries(listToSeries(C, "C(t)", T));
    dataset.addSeries(listToSeries(I, "I(t)", T));

    JFreeChart chart = ChartFactory.createXYLineChart("Графики Y(t), C(t), I(t)", "t", "", dataset);
    chart.getXYPlot().getDomainAxis().setRange(new Range(1, T));
    File target = new File("result.png");
    try {
        ChartUtilities.saveChartAsPNG(target, chart, 450, 400);
        Desktop.getDesktop().open(target);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

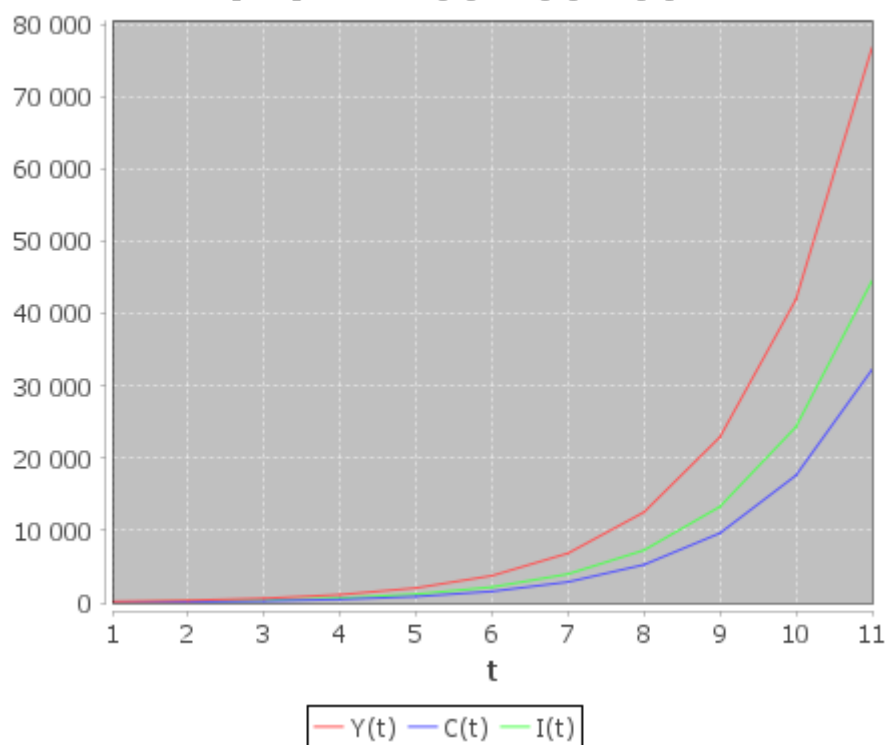
private List<Double> initWithZeroes(int capacity) {
    List<Double> list = new ArrayList<>{Arrays.asList(new Double[capacity+2])};
    Collections.fill(list, (double)0);
    return list;
}

private XYSeries listToSeries(List<Double> values, String name, int size) {
    XYSeries series = new XYSeries(name);
    for(int i = 1; i <= size; i++) {
        series.add(i, values.get(i));
    }
    return series;
}
}

```

Результат работы:

### Графики $Y(t)$ , $C(t)$ , $I(t)$



Вывод: в результате работы была изучена модель Харрода – Домара, написана программа для расчета объема производства и построения графиков производства, потребления и инвестиций.