package konvul\_odev;

import java.util.Scanner;

import org.knowm.xchart.CategoryChart;

import org.knowm.xchart.CategoryChartBuilder;

import org.knowm.xchart.CategorySeries.CategorySeriesRenderStyle;

import org.knowm.xchart.SwingWrapper;

import org.knowm.xchart.style.Styler.LegendPosition;

import org.knowm.xchart.QuickChart;

import org.knowm.xchart.SwingWrapper;

import org.knowm.xchart.XYChart;

public class Konvul\_Odev {

public static void main(String[] args) {

Scanner scn = new Scanner(System.in);

System.out.println("Yapmak İstediğiniz İslemi Önündeki Numaraları Tuşlayarak Yapabilirsiniz.....");

System.out.println(" 2) X[n] ve H[n] Grafiklerinin (Giris) Ayrık Zamanlı Sinyalleri için 2'ye Basınız... ");

System.out.println(" 1) Y[n] Grafiğinin (Sonuc)Ayrık Zamanlı Sinyali için 1'e Basınız...");

int secim = scn.nextInt();

if (secim == 1) {

ExampleChart<CategoryChart> konvuluChart = new KonvulGraf();

CategoryChart chart = konvuluChart.Konvulusyon();

new SwingWrapper<CategoryChart>(chart).displayChart();

} else if (secim == 2) {

System.out.println("Lütfen Değerleri Doğru Giriniz:: ");

// Javadaki xChart kütüphanesini kullanarak Ekrana Grafiği yazdırdım.

System.out.println("1) X[n] Grafiği İçin 1'e Basınız...");

System.out.println("2) H[n] Grafiği İçin 2'e Basınız...");

int secim2=scn.nextInt();

if (secim2==1) {

Chart<CategoryChart> exampleChart1 = (Chart<CategoryChart>) new x();

CategoryChart chart1 = exampleChart1.getChart();

new SwingWrapper<>(chart1).displayChart();

}

else if (secim2==2) {

Chart<CategoryChart> exampleChart2 = (Chart<CategoryChart>) new h();

CategoryChart chart2 = exampleChart2.getChart();

new SwingWrapper<>(chart2).displayChart();

}

else{

System.out.println("Hatalı Tuşlama");

}

} else {

System.out.println("Yanlış Tuşlama Yaptınız.");

}

}

}

class KonvulGraf implements ExampleChart<CategoryChart> {

public CategoryChart Konvulusyon() {

Scanner scn = new Scanner(System.in);

//Burada x grafiği x(yani yatay) nerden başladığına göre işlem yaptığımız için başlangıç değerini kullanıcıdan alırız

System.out.print("X[n] Grafiğinin başlama noktasını Giriniz : ");

int xBasla = scn.nextInt();

//Burada y grafiği x(yani yatay) değerleinin nerden başladığına göre işlem yaptığımız için başlangıç değerini kullanıcıdan alırız

System.out.print("H[n] Grafiğinin başlama noktasını Giriniz : ");

int hBasla = scn.nextInt();

//Kullanıcıdan kaç tane değer giriceğini sormamızın amaçı dizimiz kaç boyutlu olacağını öğrenmemizdir

System.out.print("X Grafiğinin Grafik Uzunlukunluğunu Giriniz : ");

int Xlenght = scn.nextInt();

System.out.print("H Grafiğinin Grafik Uzunlukunluğunu Giriniz : ");

int Hlenght = scn.nextInt();

int x[];// X Grafiğinin Değerleri

int h[];// Y Grafiğinin Değerleri

//Burada eleman sayısına göre islem yapacağız yani daha çok olan eleman sayısına göre işlem yapıcağız

if (Xlenght >= Hlenght) {

h = new int[Xlenght];

x = new int[Xlenght];

} else {

h = new int[Hlenght];

x = new int[Hlenght];

}

//Grafiklerin başlangıç değerlerinden itibaren X eksenine karşılık Gelen Y değerleri

System.out.print("Toplamda " + Xlenght + " tane X Grafiğinin Değeri Vardır ve İstenilen İndise Göre Giriniz \n"

+ "(Yani Girdiğiniz Başlangıç Değerinden İtibaren x-eksenine Karşılık Gelen Değerleri Yazınız....)\n");

for (int i = 0; i < Xlenght; i++) {

System.out.println(xBasla + " . Değerinin X Grafiğindeki Y-eksenine Karşılık Gelen Sayıyı Giriniz: \n");

x[i] = scn.nextInt();

xBasla++;

}

System.out.print("Toplamda " + Hlenght + " tane X Grafiğinin Değeri Vardır ve İstenilen İndise Göre Giriniz \n"

+ "+ \"(Yani Girdiğiniz Başlangıç Değerinden İtibaren x-eksenine Karşılık Gelen Değerleri Yazınız....)\n");

for (int i = 0; i < Hlenght; i++) {

System.out.println(hBasla + " . Değerinin H Grafiğindeki Y-eksenine Karşılık Gelen Sayıyı Giriniz: \n");

h[i] = scn.nextInt();

hBasla++;

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Grafik Tablosu Açılmıştır...");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//x grafiği

int diziXy[]=new int[Xlenght];

int diziX[]=new int[Xlenght];

for (int i = 0; i < Xlenght; i++) {

diziX[i]=x[i];

diziXy[i]=xBasla;

xBasla++;

}

//h grafiği

int diziHy[]=new int[Hlenght];

int diziH[]=new int[Hlenght];

for (int i = 0; i < Hlenght; i++) {

diziH[i]=x[i];

diziHy[i]=xBasla;

hBasla++;

}

//bundan sonra girilen değerli tabloları yani h[n] ve x[n] tablolarını ekrana çıktı olarak verme

//'GrafBasla' değişkeni Grafiğin başlangıç değerini bulur yani sonucun nereden başladığını gösterir

int GrafBasla = xBasla + hBasla;

//Grafiğin Y ekseninde kaç tane değer olduğunu gösteren değerdir 'GrafUzunluk' değeri

int GrafUzunluk = (x.length + h.length) - 1;

int xEksen[] = new int[GrafUzunluk];

//Bu for döngüsünde sonuç grafiğinin x eksenin başlangıç değerlerini ve son değere kadar bilgileri Eksen dizisine atar

for (int i = 0; i < GrafUzunluk; i++) {

xEksen[i] = GrafBasla;

GrafBasla++;

}

//yEksenin uzunluğunu yukarıdan ,girilen x grafiğinin ve h grafiğinin boyutunu toplamı -1 kadar olduğunu gözlemlerimde çıkardım

int yEksen[] = new int[GrafUzunluk];

int indis = 0;

//GrafUzunluğunun yarısının +1 i kadar for dönüyor

for (int i = 0; i < GrafUzunluk / 2 + 1; i++) {

//i kadar döner teker teker çarpıyoruz değerleri

for (int j = 0; j <= i; j++) {

yEksen[indis] += h[i - j] \* x[j];

}

indis++;

}

int indis2 = indis;

for (int i = indis - 1; i > 0; i--) {

for (int j = 1; j <= i; j++) {

//Konvülüsyon İşleminin algoritmik gösterimi

yEksen[indis2] += h[indis - j] \* x[indis - i + j - 1];

}

indis2++;

}

//Burada da Java'nın xChart kütüphanesinden faydalanarak Grafiğimizin Ayrık zamanda Karşılığını yazdırabildik

//Bu satırda açılan pencerenin boyutunu ayarlayabiliriz

CategoryChart chart = new CategoryChartBuilder().width(1000).height(1000).title("GRAFİK TABLOSU").build();

//Burada Grafik Penceresinin ekranda gözükmesini ayarlıyoruz

chart.getStyler().setChartTitleVisible(true);

chart.getStyler().setLegendPosition(LegendPosition.InsideNW);

chart.getStyler().setDefaultSeriesRenderStyle(CategorySeriesRenderStyle.Stick);

chart.addSeries("Y[n]", xEksen, yEksen);

// chart.addSeries("X[n]", diziX, diziXy);

//chart.addSeries("H[n]", diziH, diziHy);

return chart;

}

//Burda Giriş Sinyali Olan X[n] Sinyalini Ekrana xChart vasıtasıyla çıkaracağız

}

class h implements Chart<CategoryChart> {

Scanner s = new Scanner(System.in);

static int hbas;

static int huz;

static int h[];

@Override

public CategoryChart getChart() {

System.out.print("h[n] grafiğinin başlama noktasını Giriniz: ");

hbas = s.nextInt();

System.out.print("H[n] Grafiğinin Uzunluğunu Giriniz: ");

huz = s.nextInt();

System.out.print(huz + " Tane H[n] değerlerini Giriniz..: ");

h = new int[huz];

for (int i = 0; i < huz; i++) {

System.out.println("H[n] Grafiğinin Değerini Giriniz:");

h[i] = s.nextInt();

}

int Datay[] = new int[huz];

for (int i = 0; i < huz; i++) {

Datay[i] = hbas;

hbas++;

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Grafik Tablosu Açılmıştır...");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

CategoryChart chart = new CategoryChartBuilder().width(800).height(600).title("Stick").build();

chart.getStyler().setChartTitleVisible(true);

chart.getStyler().setLegendPosition(LegendPosition.InsideNW);

chart.getStyler().setDefaultSeriesRenderStyle(CategorySeriesRenderStyle.Stick);

chart.addSeries("h[n]", Datay, h);

return chart;

}

}

class x implements Chart<CategoryChart> {

Scanner s = new Scanner(System.in);

static int xbas;

static int xuz;

static int x[];

@Override

public CategoryChart getChart() {

System.out.print("X[n] Grafiğinin Başlama Noktasını Giriniz: ");

xbas = s.nextInt();

System.out.print("X[n] Grafiğinin Uzunluğunu Giriniz: ");

xuz = s.nextInt();

System.out.print(xuz + " Tane X[n] Değerlerini Giriniz...\n");

x = new int[xuz];

for (int i = 0; i < xuz; i++) {

System.out.println("X[n] Grafiğinin Değerini Giriniz:");

x[i] = s.nextInt();

}

int Datax[] = new int[xuz];

for (int i = 0; i < xuz; i++) {

Datax[i] = xbas;

xbas++;

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Grafik Tablosu Açılmıştır...");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

CategoryChart chart = new CategoryChartBuilder().width(800).height(600).title("Stick").build();

chart.getStyler().setChartTitleVisible(true);

chart.getStyler().setLegendPosition(LegendPosition.InsideNW);

chart.getStyler().setDefaultSeriesRenderStyle(CategorySeriesRenderStyle.Stick);

chart.addSeries("x[n]", Datax, x);

return chart;

}

}

interface Chart<T> {

public CategoryChart getChart();

}