package chartproject;

import java.awt.\*;

import java.awt.Event.\*;

import jpb.DrawableFrame;

public class Main {

public static void main(String[] args) {

if (args.length == 0)

System.out.println("Please enter arguments");//checks to make sure that there are arguments

else

System.out.println();

int width = 0;

int height = 0;

int graphArray[] = new int [args.length];//sets the array of bars to the length of entered arguments

for (int i = 0; i<args.length; i++){//

int currentValue = Integer.parseInt(args[i]);//gets the value in the args array and converts it from a string to an integer

graphArray[i] = currentValue;//takes the converted value and stores it in the graphArray

}

for (int i = 0; i<graphArray.length; i++){ //just made a loop to print out the values in the array

System.out.println(graphArray[i]);

}

width = graphArray[0]; //assigns the first parameter to width

height = graphArray[1]; //assigns the second parameter to height

DrawableFrame df = new DrawableFrame("Chart"); //made a window named "chart"

df.setVisible(true); //set the window to visible

df.setSize(width, height); //set the window to take the arguments of width, and height

int max=0;

for (int i = 2; i<graphArray.length; i++){ //excludes the first two arguments of width and height and compares the others

if(max<graphArray[i]) //the highest argument is set to max

max = graphArray[i] + 20; //20 is added to it so that the tallest bar will be 20 pixels from the top

}

int spaces= 0; //spaces between each bar

int z = 0;

int widthOfEachBar = 0;

spaces = (graphArray.length-1)\*5; //-1 because graphArray.length will have 2 more

elements than needed b/c of the width and height parameters.

//and there is one more space than there are bars. x5 because it will tell you how much space is between the bars in total.

z = width-spaces; //when you do this it tells you how much space there is after you subtract the spaces between each bar

widthOfEachBar = z; /(graphArray.length-2);//tells you the width of each bar

for (int i= 2; i < 3; i++){

Graphics[] rectangleArray = new Graphics[graphArray.length];//the loop for the first bar

rectangleArray[2] = df.getGraphicsContext(); //it's sepaerate from the rest b/c there the first bar needs to be 5 from the left

int x = 5; //a different formula is used for the other bars

rectangleArray[2].fillRect(x, (max-graphArray[2]), widthOfEachBar, graphArray[2]);

}

int redComp = 255; //color is divided into red, green, and blue.

int greenComp = 127; //here im setting their colors so that they can be changed by number later

int blueComp = 0;

int j= 1; //j is the current bar

int s = 10; //space after each bar

for (int i = 3; i < graphArray.length; i++){

Graphics[] rectangleArray = new Graphics[graphArray.length];

rectangleArray[i] = df.getGraphicsContext();

int x = widthOfEachBar \* j + s; //multiplies the barWidth by the bar #,(j which is incremented each iteration), then adds ten for spacing

rectangleArray[i].setColor(new Color(redComp, greenComp, blueComp));

rectangleArray[i].fillRect(x, max-graphArray[i], widthOfEachBar, graphArray[i]);//applies the colors to the bars

redComp= redComp-30;

blueComp= blueComp+30; //changes the colors

greenComp = greenComp +5;

j++;

s+=5;

}

}

}