Christopher Bussen, Muhammad Alsarraf, Khaled Almutairi

CPS 150 02 – Algorithms and Programming 1

Lab Group Project 2

9/17/20

**Program 1 Algorithm**

1. Start program
2. Import the JOption class
3. Prompt the user to enter the total width of the space
4. Declare a double variable to store the input for total width of the space
5. Prompt the user to enter the width of each tile
6. Declare a double variable to enter the width of each tile
7. Declare an int variable to store the number of groups of tiles – equal to (total width – tile width) / (4 \* tile width)
8. Declare an int variable to store the total number of tiles – equal to (number of groups \* 4) + 1
9. Declare an int variable to store the number of black tiles – equal to number of groups + 1
10. Declare an int variable to store the number of white tiles – equal to number of groups
11. Declare an int variable to store the number of gray tiles – equal to number of groups \* 2
12. Declare a double variable to store the width of the gap on each side – equal to (total width – (tile width \* number of tiles)) / 2
13. Print the total number of tiles, number of black, white, and gray tiles, and the width of the gap on each side
14. End program

**Program 1 Running Screenshots**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Program 1 Code**

/\*

Christopher Bussen

CPS 150 02

Lab Group Project 2

BlackAndWhiteTiles: number, number; number, number, number, number, number

program takes in a value from the user for the total width of the space and tile width,

and calculates the total number of black, white, and gray tiles, and the width of the gap

on each side

# of groups = (total width - tile width) / (4 \* tile width)

# of tiles = (# of groups \* 4) + 1

# of black = # of groups + 1 -- one black per group and need one at the end

# of white = # of groups -- one white per group

# of gray = # of groups \* 2 -- two grays per group

ex1: user inputs 150, 5 program outputs 29, 8, 7, 14, 2.5

ex2: user inputs 212, 8.5 - program outputs 21, 6, 5, 10, 16.75

ex3: user inputs 42391, 5000.5 - program outputs 5, 2, 1, 2, 8694.25

ex4: user inputs five hundred, x - program outputs error

ex5: user inputs -412, 6.25 - program outputs -63, -15, -16, -32, -9.125 -- doesn't make sense

because number is negative

\*/

import javax.swing.JOptionPane;

public class BlackAndWhiteTiles {

public static void main(String [] args){

//Step 1: import JOption class and prompt user to enter total width and tile width

String input1 = JOptionPane.showInputDialog("Enter the total width: ");

double totalWidth = Double.parseDouble(input1);

String input2 = JOptionPane.showInputDialog("Enter the tile width: ");

double tileWidth = Double.parseDouble(input2);

//Step 2: calculate number of groups

int numGroups = (int) ((totalWidth - tileWidth) / (4 \* tileWidth));

//Step 3: calculate number of tiles

int numTiles = 1 + (numGroups \* 4);

//Step 4: calculate the number of black and white tiles

int numBlack = numGroups + 1;

int numWhite = numGroups;

int numGray = 2 \* numGroups;

//Step 5: calculate the gap

double gap = (totalWidth - (tileWidth \* numTiles)) / 2;

//Step 6: show results using dialog box

JOptionPane.showMessageDialog(null, "Total number of tiles: " + numTiles + "\n Total number of black tiles: " + numBlack + "\n Total number of white tiles: " + numWhite + "\n Total number of gray tiles: " + numGray + "\n Gap on each side should be: " + gap + "inches");

}

}

**Program 2 Algorithm**

1. Start program
2. Import scanner
3. Take the input from user for number of rows in rows.
4. Take the input from user for number of columns in cols.
5. Loop rows number of times
6. Loop Cols number of times
7. Check if rows is even and cols is odd then print 1
8. If not, check if rows is odd and cols is even then print 1
9. Otherwise print 0
10. End loop
11. Print new line
12. End loop
13. End program

**Program 2 Running Screenshot**

A screenshot of a computer

Description automatically generated

**Program 2 Code**

import java.util.\*;

public class Problem2 {

public static void main(String ar[])

{

Scanner sc=new Scanner(System.in);

int rows=sc.nextInt();

int cols=sc.nextInt();

for(int i=1;i<=rows;i++)

{

for(int j=1;j<=cols;j++)

{

if( i%2==0 && j%2!=0)

System.out.print(" 1 ");

else if( i%2!=0 && j%2==0)

System.out.print(" 1 ");

else

System.out.print(" 0 ");

}

System.out.println();

}

}

}