/\*

\*Title: lost in the forest simulator

\* File: forest.java

\* Programmers:steven guo

\* Email:sgpgk@mail.umsl.edu

\* Course:4500-001

\* Date: 2/4/20

\* Summary:this program simulates two people in a forest. The simulation ends when the two people

\* meet in the forest or when the time of each person reaches 1000000

\*

\*/

import java.util.Scanner;

import java.util.Random;

public class forest

{

static int a; //this int is the length of the forest

static int b; //this int is the height of the forest

static int pat\_x; //this int is the x position of patrick in the forest

static int pat\_y; //this int is the y position of patrick in the forest

static int chris\_x; //this int is the x position of chris in the forest

static int chris\_y; //this int is the y position of chris in the forest

static int chrisSteps; //this int is how many times chris moves in the forest

static int patSteps; //this int is how many times pat moves in the forest

static int time; //this int keeps track of how many moves each person makes

//this method sets pat to the upper left and chris to bottom right

public static void setPeople()

{

pat\_x=0;

pat\_y=b;

chris\_x=a;

chris\_y=0;

}

//this method takes in int person which determines who is moving and int direction determines what direction the person is moving

public static void move(int person, int direction)

{

time++; //time is incremented because a person is moving

if(person ==1) //if person is 1 then it is pat that is moving, when person is anything else then its chris that is moving

{

patSteps++;

switch(direction) //this switch case statement moves the person based on case 1-8 which contains all directions that chris and pat move

{

case 1: //this moves the person north

if(pat\_y+1 <= b)//checks to see if the movement will cause the person to go out of bounds

{

pat\_y++; //the y position is incremented to move the person

}

break;

case 2: //this moves the person northeast

if(pat\_x+1 <= a && pat\_y+1 <= b)

{

pat\_x++;

pat\_y++;

}

break;

case 3: //this moves the person east

if(pat\_x+1 <= a)

{

pat\_x++;

}

break;

case 4: //this moves the person southeast

if(pat\_x+1 <= a && pat\_y-1 >=0)

{

pat\_x++;

pat\_y--;

}

break;

case 5: //this moves the person south

if(pat\_y-1 >=0)

{

pat\_y--;

}

break;

case 6: //this moves the person southwest

if(pat\_x-1 >= 0 && pat\_y-1 >= 0)

{

pat\_x--;

pat\_y--;

}

break;

case 7: //this moves the person west

if(pat\_x-1 >= 0)

{

pat\_x--;

}

break;

case 8: //this moves the person northwest

if(pat\_x-1 >= 0 && pat\_y+1 <=b)

{

pat\_x--;

pat\_y++;

}

break;

default:

}

}

else

{

chrisSteps++;

switch(direction)

{

case 1:

if(chris\_y+1 <= b)

{

chris\_y++;

}

break;

case 2:

if(chris\_x+1 <= a && chris\_y+1 <= b)

{

chris\_x++;

chris\_y++;

}

break;

case 3:

if(chris\_x+1 <= a)

{

chris\_x++;

}

break;

case 4:

if(chris\_x+1 <= a && chris\_y-1 >=0)

{

chris\_x++;

chris\_y--;

}

break;

case 5:

if(chris\_y-1 >=0)

{

chris\_y--;

}

break;

case 6:

if(chris\_x-1 >= 0 && chris\_y-1 >= 0)

{

chris\_x--;

chris\_y--;

}

break;

case 7:

if(chris\_x-1 >= 0)

{

chris\_x--;

}

break;

case 8:

if(chris\_x-1 >= 0 && chris\_y+1 <=b)

{

chris\_x--;

chris\_y++;

}

break;

default:

}

}

}

//this method checks to see if the halting conditions are met it returns 1,2 based on how the simulation ends so i can output to the user on what happened.

//returns 0 when the simulation continues

public static int turnCheck()

{

if(time== 100000) //checks if 1 million turns have gone by

{

return 1;

}

else if(pat\_x == chris\_x && pat\_y == chris\_y) //checks if pat and chris meet

{

return 2;

}

else

{

return 0;

}

}

public static void printForest(int num)

{

switch(num)

{

case 1:

System.out.println("The simulation ended with 1000000 movements made");

break;

case 2:

System.out.println("The simulation ended with pat and chris meeting");

System.out.println("pat moved " + patSteps + " times");

System.out.println("chris moved " + chrisSteps + " times");

System.out.println("Total turns: " + time);

}

}

public static void main(String[] args)

{

Random rand = new Random();

Scanner input = new Scanner(System.in);

int movement; //this int grabs the random integer 1-8 that determines the direction the person will move

System.out.println("In this simulation there are two people that wander around the woods until they meet. You get to decide how big the woods are! Please enter the dimensions for the woods.");

System.out.println("Enter the length of the woods. Enter an integer that is greater than or equal to 2 and less than or equal to 50.");

a=input.nextInt(); //this sets the length of the barrier

if(a < 2 || a > 50)

{

//this do-while loop makes the user enter the right user input for simulation

do

{

System.out.println("ERROR! Length requirements not met. Enter an integer from 2-50.");

a=input.nextInt();

}

while(a < 2 || a > 50); //this loop keeps going when user input is incorrect

}

System.out.println("Enter the height of the woods. Enter an integer that is greater than or equal to 2 and less than or equal to 50");

b = input.nextInt();

if(b < 2 || b > 50)

{

do

{

System.out.println("ERROR! Height requirements not met. Enter an integer that is greater than or equal to 2 and less than or equal to 50.");

b=input.nextInt(); //this sets the height of the barrier

}

while(b < 2 || b > 50);

}

setPeople();//this initializes pat and chris into their starting spots

do

{

movement = rand.nextInt(8) + 1; //random integer for direction

move(1,movement); //moves pat in that direction

if(turnCheck()==1) //I have this if else statement here so that i can check for halt conditions after pat moves

{

printForest(1); //prints out the results of the simulation

break; //breaks out of while loop so that it won't go to chris's movement

}

else if(turnCheck()==2)

{

printForest(2);

break;

}

else

{

}

movement = rand.nextInt(8) + 1;

move(2,movement);

if(turnCheck()==1)

{

printForest(1);

break;

}

else if(turnCheck()==2)

{

printForest(2);

break;

}

else

{

}

}

while(turnCheck()==0);

}

}