Chapter 3: Variables (Solutions)

Solution to Task 21:

public void act() {

 int count = 0;

 while (!treeFront()) {

 move();

 if (onLeaf()) {

 count = count + 1;

 }

 }

 System.out.println("The result is: " + count);

 stop();

}

Solution to Task 22:

public class MyKara extends Kara {

 boolean goingRight = true;

 public void act() {

 invertField();

 if (treeFront()) {

 if (goingRight) {

 // we are at the right border

 turnAroundRight();

 } else {

 // we are at the left border

 turnAroundLeft();

 }

 } else {

 move();

 }

 }

 public void turnAroundRight() {

 if (treeRight()) {

 // we are in the bottom right corner

 stop();

 } else {

 turnRight();

 move();

 turnRight();

 goingRight = false;

 }

 }

 public void turnAroundLeft() {

 if (treeLeft()) {

 // we are in the bottom left corner

 stop();

 } else {

 turnLeft();

 move();

 turnLeft();

 goingRight = true;

 }

 }

 public void invertField() {

 if (onLeaf()) {

 removeLeaf();

 } else {

 putLeaf();

 }

 }

}

Solution to Task 23:

public class MyKara extends Kara {

 boolean goingRight = true;

 int step = 0;

 public void act() {

 putLeafIfEvenStep();

 if (treeFront()) {

 if (goingRight) {

 // we are at the right border

 turnAroundRight();

 } else {

 // we are at the left border

 turnAroundLeft();

 }

 } else {

 move();

 step = step + 1;

 }

 }

 public void turnAroundRight() {

 if (treeRight()) {

 // we are in the bottom right corner

 stop();

 } else {

 turnRight();

 move();

 turnRight();

 goingRight = false;

 step = step + 1;

 }

 }

 public void turnAroundLeft() {

 if (treeLeft()) {

 // we are in the bottom left corner

 stop();

 } else {

 turnLeft();

 move();

 turnLeft();

 goingRight = true;

 step = step + 1;

 }

 }

 public void putLeafIfEvenStep() {

 if (step % 2 == 0) {

 // even step number --> put a leaf

 putLeaf();

 }

 }

}

Solution to Task 24:

public class MyKara extends Kara {

 int longestRow = 0;

 public void act() {

 while (!onLeaf()) {

 if (treeFront()) {

 countRow();

 } else {

 move();

 }

 }

 System.out.println("The longest tree line is " + longestRow

 " trees long");

 stop();

 }

 public void countRow() {

 int currentRow = 0;

 turnLeft();

 while (treeRight()) {

 currentRow = currentRow + 1;

 move();

 }

 // go around tree line

 turnRight();

 move();

 move();

 turnRight();

 // go back down

 int i = 0;

 while (i < currentRow) {

 move();

 i = i + 1;

 }

 turnLeft();

 // test whether the current row is longer

 if (currentRow > longestRow) {

 longestRow = currentRow;

 }

 }

}