

8.8

代码：

**import** java.util.Scanner;

**public** **class** one {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of points: ");

**int** number = input.nextInt();

**double**[][] points = **new** **double**[number][2];

System.***out***.print("Enter " + number + " points: ");

**int** length=points.length;

**for** (**int** i = 0; i <length; i++) {

points[i][0] = input.nextDouble();

points[i][1] = input.nextDouble();

}

**int**[] p1=**new** **int**[length];

**int**[] p2=**new** **int**[length];

p1[0]=0;p2[0]=1;

**int** n=0,m=0;

**double** shortest = *distance*(points[p1[0]][0], points[p1[0]][1], points[p2[0]][0], points[p2[0]][1]);

**for** (**int** i = 0; i < points.length; i++) {

**for** (**int** j = i + 1; j < points.length; j++) {

**double** distance = *distance*(points[i][0], points[i][1],points[j][0], points[j][1]);

**if** (shortest >= distance) {

p1[n++] = i;

p2[m++] = j;

shortest = distance;

}

}

}

**for**(**int** i=0;i<length;i++)

{

**if**(points[p1[i]][0]!=points[p2[i]][0]&&points[p1[i]][1]!=points[p2[i]][1])

{

System.***out***.println("The closest two points are " + "(" + points[p1[i]][0] + ", " + points[p1[i]][1] + ") and (" + points[p2[i]][0] + ", " + points[p2[i]][1] + ")");

}

}

System.***out***.println("Their distance is "+shortest);

}

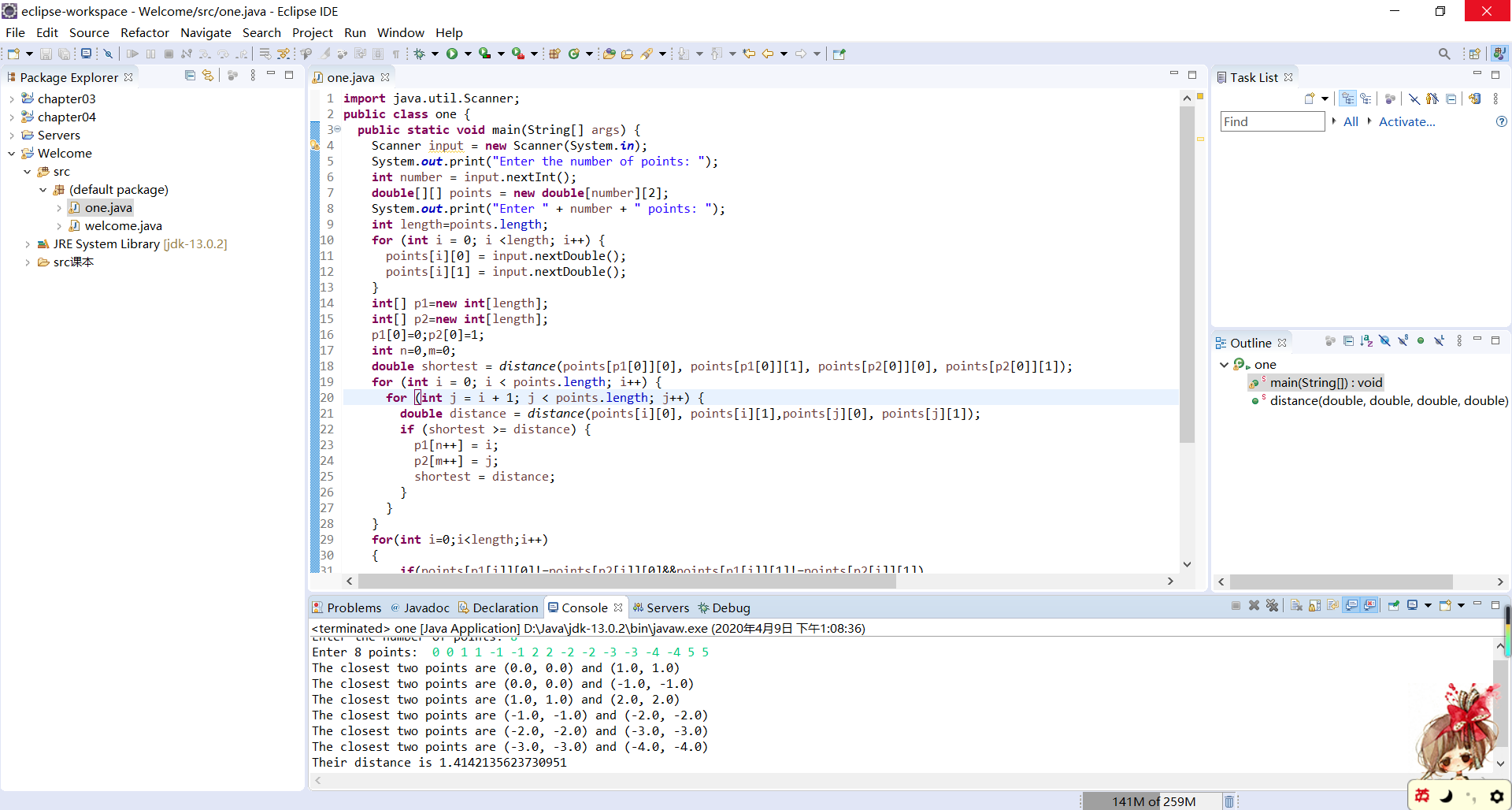
**public** **static** **double** distance( **double** x1, **double** y1, **double** x2, **double** y2) {

**return** Math.*sqrt*((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

}

}

截图：



8.9

代码：

**import** java.util.Scanner;

**public** **class** two {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

**int** r1, c1, r2, c2;

**char**[][] play = **new** **char**[3][3];

**for** (**int** i = 0; i < 3; i++) //初始化棋盘

{

**for** (**int** j = 0; j < 3; j++) {

play[i][j] = ' ';

}

}

*show*(play);

**while** (*check*(play)) {

System.***out***.println("Enter a row (0,1, or 2) for player X: ");

r1 = input.nextInt();

System.***out***.println("Enter a column (0,1, or 2) for player X: ");

c1 = input.nextInt();

play[r1][c1] = 'X';

*show*(play);

**if** (*check*(play)) {

System.***out***.println("Enter a row (0,1, or 2) for player O: ");

r2 = input.nextInt();

System.***out***.println("Enter a column (0,1, or 2) for player O: ");

c2 = input.nextInt();

play[r2][c2] = 'O';

*show*(play);

}

**else** **break**;

}

}

**public** **static** **void** show(**char**[][] play)// 展示棋盘

{

**for** (**int** i = 0; i < 3; i++) {

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

**for** (**int** j = 0; j < 3; j++) {

System.***out***.print("|");

System.***out***.print(play[i][j]);

}

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

}

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

}

**public** **static** **boolean** check(**char**[][] play)// 判断输赢

{

**for** (**int** i = 0; i < 3; i++) // 同一行

{

**if** (play[i][0] == 'X' && play[i][1] == 'X' && play[i][2] == 'X') {

System.***out***.println("X player won");

**return** **false**;

} **else** **if** (play[i][0] == 'O' && play[i][1] == 'O' && play[i][2] == 'O') {

System.***out***.println("O player won");

**return** **false**;

}

}

**for** (**int** i = 0; i < 3; i++) // 同一列

{

**if** (play[0][i] == 'X' && play[1][i] == 'X' && play[2][i] == 'X') {

System.***out***.println("X player won");

**return** **false**;

} **else** **if** (play[0][i] == 'O' && play[1][i] == 'O' && play[2][i] == 'O') {

System.***out***.println("O player won");

**return** **false**;

}

}

// 同一对角线

**if** ((play[0][0] == 'X' && play[1][1] == 'X' && play[2][2] == 'X')

|| (play[0][2] == 'X' && play[1][1] == 'X' && play[2][0] == 'X')) {

System.***out***.println("X player won");

**return** **false**;

}

**if** ((play[0][0] == 'O' && play[1][1] == 'O' && play[2][2] == 'O')

|| (play[0][2] == 'O' && play[1][1] == 'O' && play[2][0] == 'O')) {

System.***out***.println("O player won");

**return** **false**;

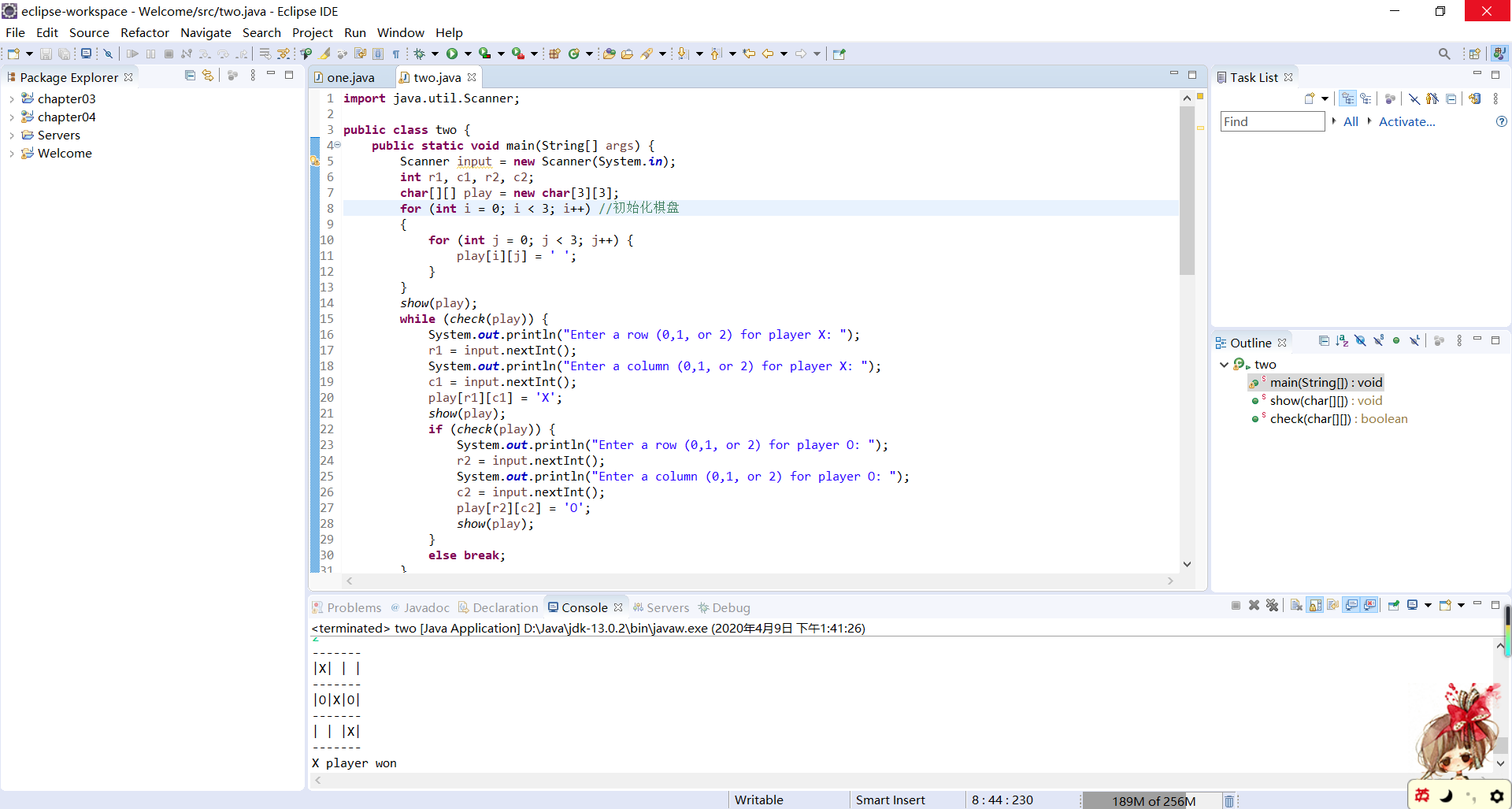
}

**return** **true**;

}

}

截图：



8.11

代码：

**import** java.util.Scanner;

**public** **class** three {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Enter a number between 0 and 511 : ");

**int** x = input.nextInt();

**int**[] answer = **new** **int**[9];

**for** (**int** i = 8; i >= 0; i--) {

**if** (x >=0 ) {

answer[i] = x % 2;

x = x / 2;

}

}

**for** (**int** i = 1; i <= 9; i++)

{

**if** (answer[i-1] == 1)

System.***out***.print("T ");

**else**

System.***out***.print("H ");

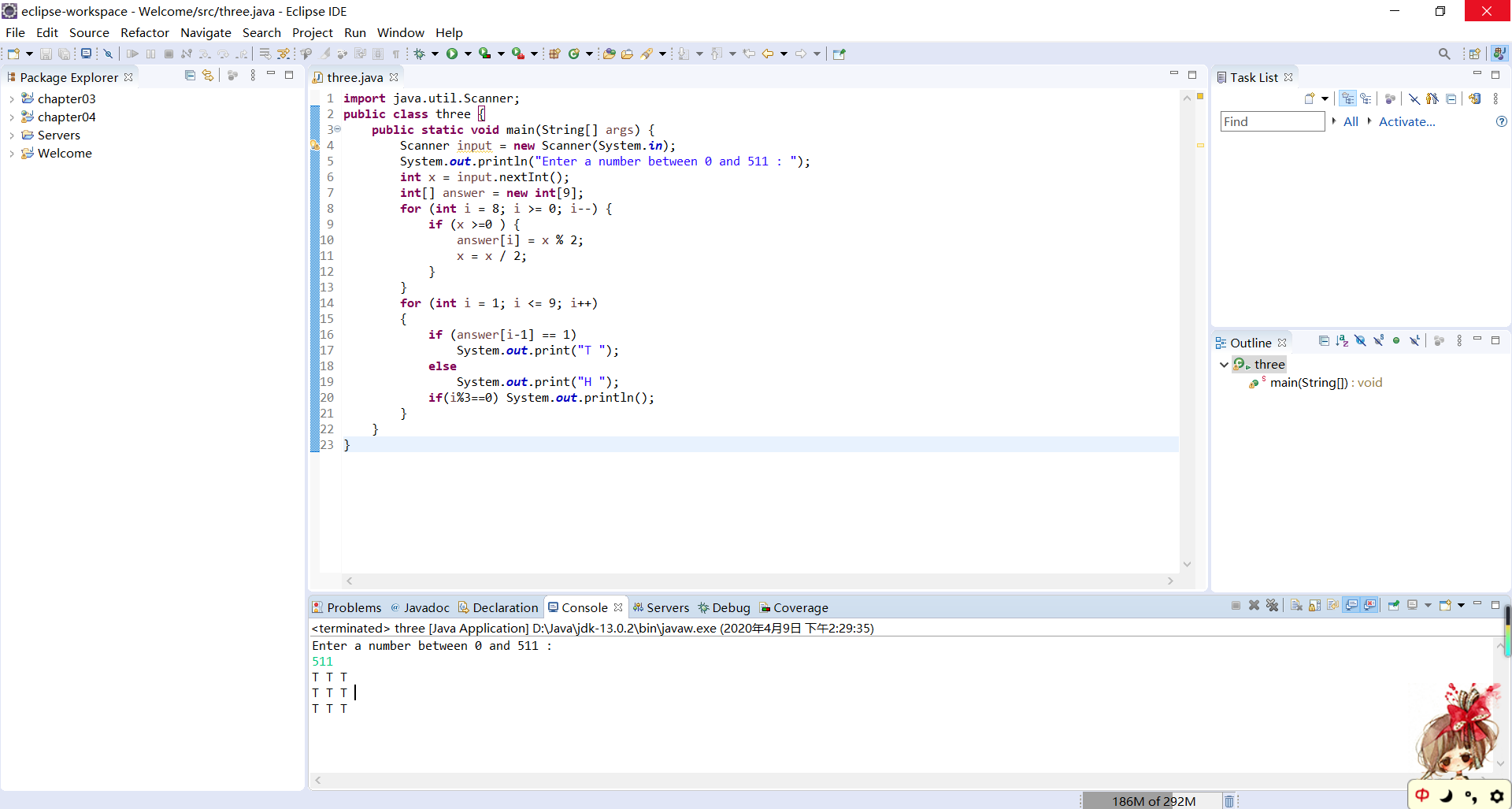
**if**(i%3==0) System.***out***.println();

}

}

}

截图：



8.19

代码：

**import** java.util.Scanner;

**public** **class** four {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Enter the row ,column and values of the two-dimensional array:");

**int** r = input.nextInt();

**int** c = input.nextInt();

**int**[][] value = **new** **int**[r][c];

**for** (**int** i = 0; i < r; i++) {

**for** (**int** j = 0; j < c; j++) {

value[i][j] = input.nextInt();

}

}

**if** (*isConsecutiveFour*(value))

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

**else**

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

}

**public** **static** **boolean** isConsecutiveFour(**int**[][] values) {

**for** (**int** i = 0; i < values.length; i++) {//行

**for** (**int** j = 0; j < values[i].length - 3; j++) {

**if**(values[i][j]==values[i][j+1]&&values[i][j]==values[i][j+2]&&values[i][j]==values[i][j+3])

{

**return** **true**;

}

}

}

**for** (**int** i = 0; i < values.length-3; i++) {//列

**for** (**int** j = 0; j < values[i].length; j++) {

**if**(values[i][j]==values[i+1][j]&&values[i][j]==values[i+2][j]&&values[i][j]==values[i+3][j])

{

**return** **true**;

}

}

}

**for** (**int** i = 0; i < values.length-3; i++) {//右斜

**for** (**int** j = 0; j < values[i].length - 3; j++) {

**if**(values[i][j]==values[i+1][j+1]&&values[i][j]==values[i+2][j+2]&&values[i][j]==values[i+3][j+3])

{

**return** **true**;

}

}

}

**for** (**int** i = 3; i < values.length; i++) {//左斜

**for** (**int** j = 3; j < values[i].length ; j++) {

**if**(values[i][j]==values[i-1][j-1]&&values[i][j]==values[i-2][j-2]&&values[i][j]==values[i-3][j-3])

{

**return** **true**;

}

}

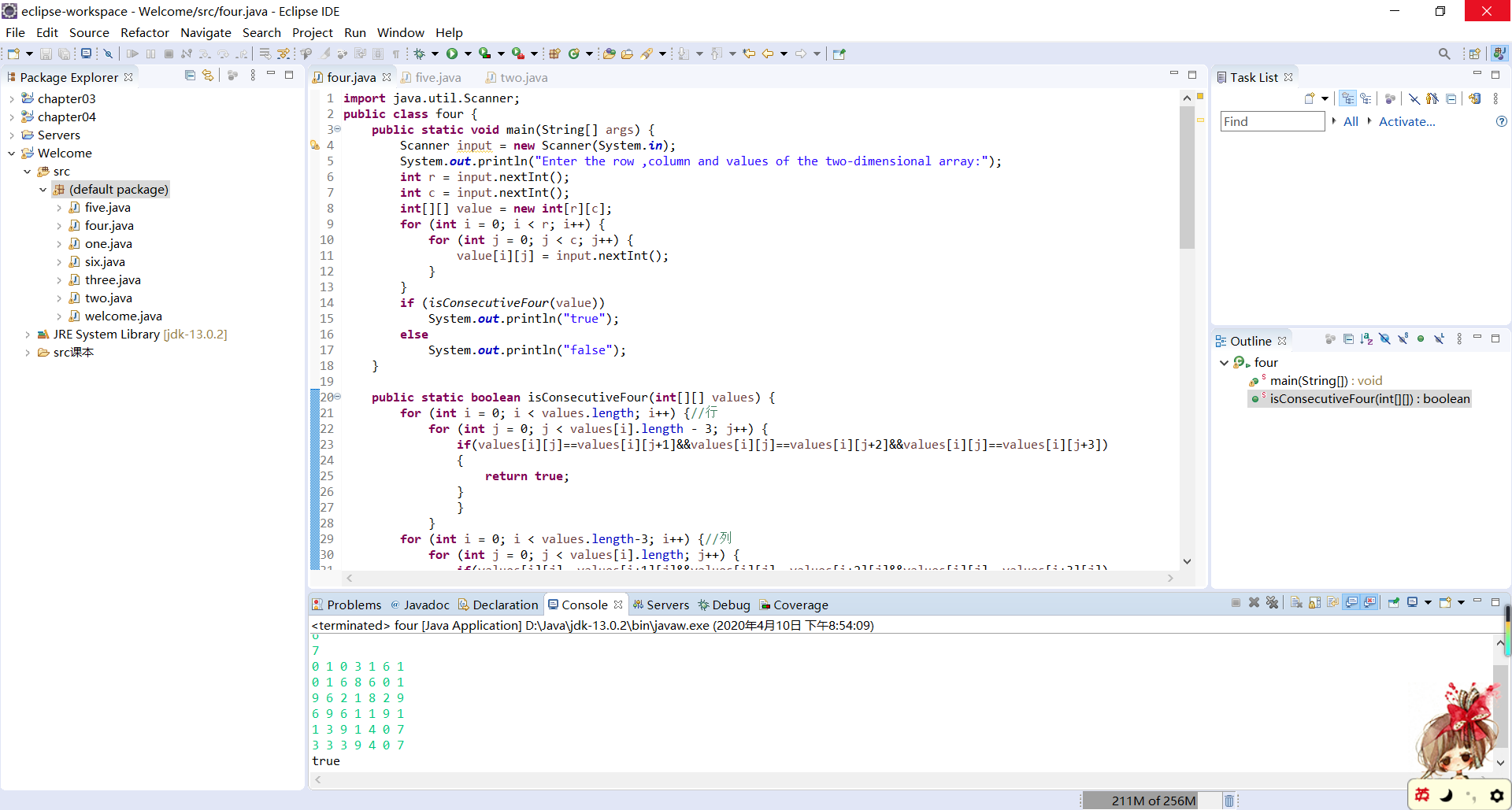
}

**return** **false**;

}

}

截图：



8.20

代码：

**import** java.util.Scanner;

**public** **class** five {

**public** **static** **void** main(String[] args) {

**char**[][] play = **new** **char**[6][7];

**for** (**int** i = 0; i < 6; i++) // 初始化棋盘

{

**for** (**int** j = 0; j < 7; j++) {

play[i][j] = ' ';

}

}

*show*(play);

**int** count;

**while** (**true**) {

*drop*("Red", play);

*show*(play);

**if** (*isWon*(play, 'R'))// R赢

{

System.***out***.println("The red player won");

**break**;

}

count = 0;// 平局

**for** (**int** i = 0; i < 6; i++) {

**for** (**int** j = 0; j < 7; j++) {

**if** (play[i][j] != ' ') {

count++;

}

}

}

**if** (count == 42) {

System.***out***.println("It ends in a draw");

**break**;

}

*drop*("Yellow", play);

*show*(play);

**if** (*isWon*(play, 'Y'))// Y赢

{

System.***out***.println("The yellow player won");

**break**;

}

count = 0;// 平局

**for** (**int** i = 0; i < 6; i++) {

**for** (**int** j = 0; j < 7; j++) {

**if** (play[i][j] != ' ') {

count++;

}

}

}

**if** (count == 42) {

System.***out***.println("It ends in a draw");

**break**;

}

}

}

**public** **static** **void** show(**char**[][] play)// 展示棋盘

{

**for** (**int** i = 0; i < 6; i++) {

**for** (**int** j = 0; j < 7; j++) {

System.***out***.print("|");

System.***out***.print(play[i][j]);

}

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

}

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

}

**public** **static** **boolean** place(**char**[][] play, **int** a, **char** x) {// 放置

**for** (**int** i = 5; i >= 0; i--) {

**if** (play[i][a] == ' ') {

play[i][a] = x;

**return** **true**;

}

}

**return** **false**;

}

**public** **static** **void** drop(String x, **char**[][] play) {// 落棋

Scanner input = **new** Scanner(System.***in***);

**boolean** done = **false**;

**do** {

System.***out***.println("Drop a " + x + " disk at column (0-6):");

**int** column = input.nextInt();

**if** (*place*(play, column, x.charAt(0)))

done = **true**;

**else**

System.***out***.println("this column is full");

} **while** (!done);

}

**public** **static** **boolean** isWon(**char**[][] play, **char** x) {

**for** (**int** i = 5; i >= 0; i--)// 行

{

**for** (**int** j = 0; j < 7 - 3; j++) {

**if** (play[i][j] == x && play[i][j + 1] == x && play[i][j + 2] == x && play[i][j + 3] == x) {

**return** **true**;

}

}

}

**for** (**int** i = 5; i >= 0 + 3; i--)// 列

{

**for** (**int** j = 0; j < 7; j++) {

**if** (play[i][j] == x && play[i - 1][j] == x && play[i - 2][j] == x && play[i - 3][j] == x) {

**return** **true**;

}

}

}

**for** (**int** i = 5; i >= 0 + 3; i--)// 左斜

{

**for** (**int** j = 0; j < 7 - 3; j++) {

**if** (play[i][j] == x && play[i - 1][j + 1] == x && play[i - 2][j + 2] == x && play[i - 3][j + 3] == x) {

**return** **true**;

}

}

}

**for** (**int** i = 5; i >= 0 + 3; i--)// 右斜

{

**for** (**int** j = 6; j >= 0 + 3; j--) {

**if** (play[i][j] == x && play[i - 1][j - 1] == x && play[i - 2][j - 2] == x && play[i - 3][j - 3] == x) {

**return** **true**;

}

}

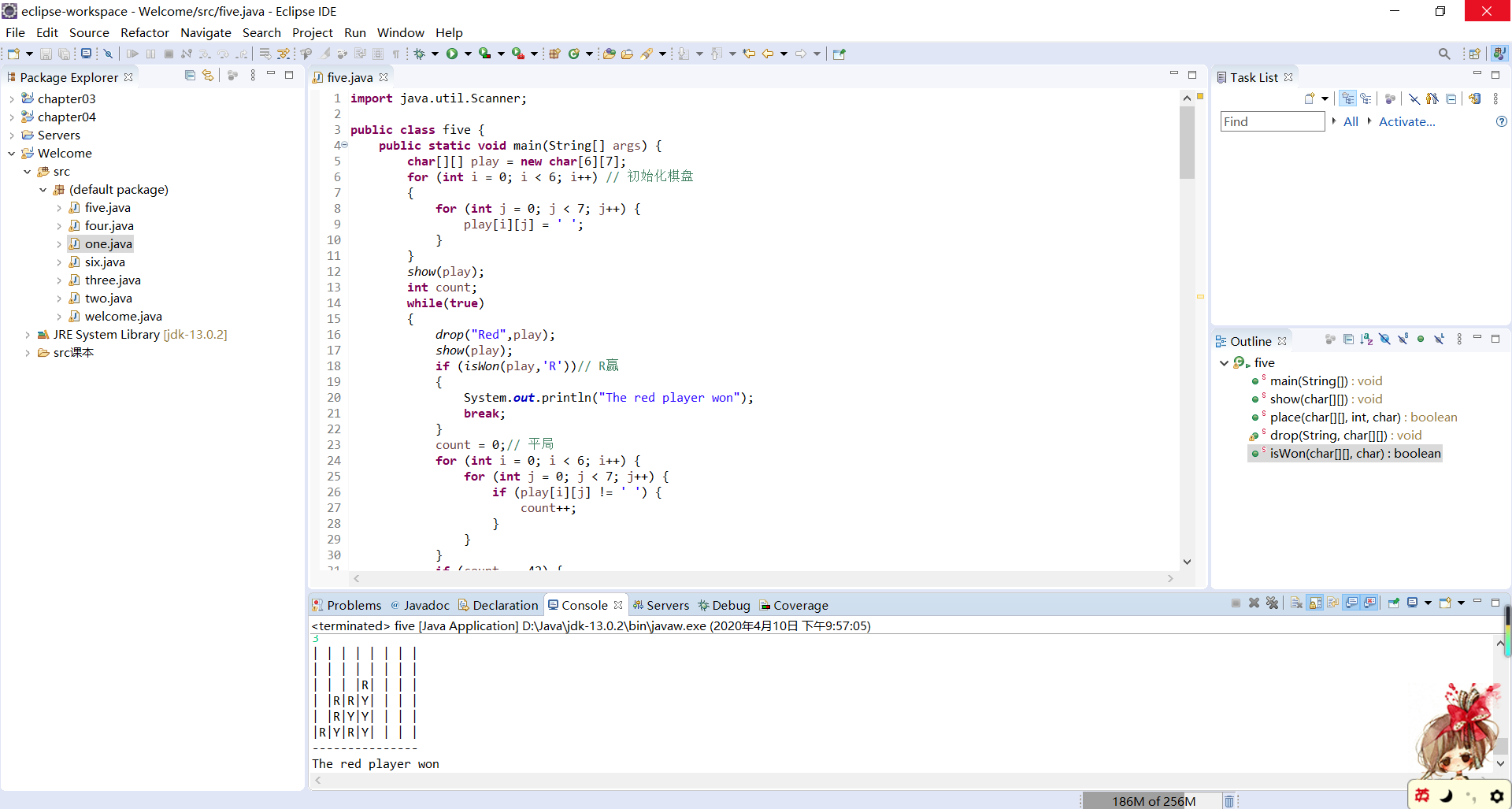
}

**return** **false**;

}

}

截图：



8.35

代码：

**import** java.util.Scanner;

**public** **class** six {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Enter the number of rows in the matrix:");

**int** row = input.nextInt();

System.***out***.println("Enter the matrix row by row:");

**int**[][] ma = **new** **int**[row][row];

**for** (**int** i = 0; i < row; i++) {

**for** (**int** j = 0; j < row; j++) {

ma[i][j] = input.nextInt();

}

}

**int** max = 0;

**int** mi = 0;

**int** mj = 0;

**for** (**int** i = 0; i < row; i++) {

**for** (**int** j = 0; j < row; j++) {

**for** (**int** k = 0; k < row - j && k < row - i; k++) {

**if** (ma[i][j] == ma[i][j + k] && ma[i][j] == ma[i + k][j] && ma[i][j] == ma[i + k][j + k]) {

**if** (max < k) {

max = k;

mi = i;

mj = j;

}

}

**else** **break**;

}

}

}

System.***out***.println("The maximum square submatrix is at ( "+mi+" , "+mj+" ) with size "+(max+1));

}

}

截图：

