**Practice «Diagonal maze»**

In the same project, take the robot out of the diagonal maze.

Start the project and explore this type of maze yourself.

Additional restrictions:

1. It is forbidden to use more than one cycle in one method.

2. It is forbidden to have methods longer than 12 lines of code.

3. It is forbidden to use the catch keyword

4. It is allowed to create helper methods, but only with friendly names, including argument names.

// Paste the final file content here DiagonalMazeTask.cs

namespace Mazes

{

public static class DiagonalMazeTask

{

public static void MoveOut(Robot robot, int width, int height)

{

int wh, hw;

wh = (width - 2) / (height - 2);

hw = (height - 2) / (width - 2);

while (robot.Finished == false)

{

if (width > height)

{ Move(robot, wh, Direction.Right, Direction.Down); }

else

{ Move(robot, hw, Direction.Down, Direction.Right); }

}

}

public static void Move(Robot robot, int aspectRatio, Direction direction, Direction directionTwo)

{

if (!robot.Finished)

for (var i = 0; i < aspectRatio; i++)

robot.MoveTo(direction);

if (!robot.Finished)

robot.MoveTo(directionTwo);

}

}

}