**Практика «Лабиринт змейка»**

В том же проекте выведите робота из лабиринта вида «змейка».

Запустите проект и самостоятельно изучите этот тип лабиринтов.

Дополнительные ограничения:

1. Запрещено использовать более одного цикла в одном методе.
2. Запрещено иметь методы длиннее 12 строк кода.
3. Запрещено использовать ключевое слово catch
4. Разрешено создавать вспомогательные методы, но только понятными именами, в том числе именами аргументов.

Обратите внимание, чтобы в вашем коде не было дублирующихся почти одинаковых методов.

// Вставьте сюда финальное содержимое файла SnakeMazeTask.cs

namespace Mazes

{

public static class SnakeMazeTask

{

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

{

int right = width - 3, down = 2;

while (robot.Finished == false)

{

MoveRight(robot, right);

MoveDown(robot, down);

MoveLeft(robot, right);

if (robot.Finished == false)

{ MoveDown(robot, down); }

}

}

public static void MoveRight (Robot robot, int a)

{

for (int i = 0; i < a; i++)

{ robot.MoveTo(Direction.Right); }

}

public static void MoveLeft(Robot robot, int a)

{

for (int i = 0; i < a; i++)

{ robot.MoveTo(Direction.Left); }

}

public static void MoveDown (Robot robot, int a)

{

for (int i = 0; i < a; i++)

{ robot.MoveTo(Direction.Down); }

}

}

}