1. Redesigned structure:
   1. Separated the labyrinth generation (InitializeLabyrinth()) and visualization (ShowLabyrinth()) in a separate class: LabyrinthBoard
   2. Separated the Top Scores logic (method ShowTopScores and string[] topScores) in a class TopScores
   3. Renamed the class Labyrinth to Engine as only the main logic of the moving remains there
2. Work on the LabyrinthBoard class:
   1. Added method bool IsPossibleCell(int row, int col) which checks if the cell on the given row and column is free to move on it and if it is within range.
   2. Added method bool isPieceOnEdge(int row, int col) which checks if the cell where the Piece is located at the moment, is on an edge
   3. Introduced constant int LabyrinthSize = 7 for the number of row and consts in the Labyrinth
   4. Introduced fields that keep the position of the moving piece (int piecePositionRow and int piecePositionCol) plus respective properties.
   5. Introduced methods that move the piece left, right up and down if possible. This is to take away this functionality from the engine.
   6. Due to the above two changes, modified the ShowLabyrinth method, so that it can print \* where the piece is;
   7. Reconfigured the ShowLabyrinth method into a ToString() method, so that the LabyrinthBoard class be decoupled from the actual means of rendering the board.
3. Work on the TopScores class:
4. Work on the Engine class:
   1. Made the engine class use the functionality of the LabyrinthBoard class:
      1. The engine doesn’t make the checks if a cell is free or not and if it possible to move there or not, instead it asks the LabyrinthBoard class
      2. The engine doesn’t keep the information where the piece is anymore, it just asks the LabyrinthBoard class where it is and to move it, if possible. Removed the fields m and n that used to keep this.
   2. Replaced the topScores functionality with the functionality from the TopScores class.
   3. Made a method “Start” which contains the logic from the Program method.