Refactoring Documentation for Project “Labyrinth”

Team "Mendelevium"

1. Redesigned the project structure:
   - Renamed the solution, project and the assembly to **Labyrinth**.
   - Renamed the main class **LabTest** to **Demo**.
   - Renamed **Labyrinth** to **Engine.**
   - Renamed **Ladder** to **Scoreboard**.
   - Renamed **Result** to **Player.**
   - Renamed **UserInputAndOutput** to **ConsoleIO.**
   - Extracted class and struct in a separate file with a good name: **Direction, Message**.

     Note: file names same as respective classes

2. Reformatted the source code:
   - StyleCop 4. 7 used to reformad code with the following warnings turned off:
     - Ordering Rules :Using Directives Must Be Placed Within NameSpace – SA1200
     - Naming Rules: Filenames Must Not Contain UnderScore(constant fields named using ALL_CAPS) – SA 1310
     - Documentation Rules: All rules turned off (XML documentation is added)
   - Removed all unnecessary empty lines and added where is needed (before each return, after each method, property, for separation of logical parts in the methods etc.).
   - Add curly brackets after each loop, conditional statement etc. Each bracket is placed alone on the line.
   - Introduced new variables to simplify complex expressions.
   - Naming Convections used: variables and fields: **camelCase**; types, methods, properties and read-only fields: **PascalCase**. Constants: **ALL_CAPS**
   - Extracted “magic” values in fields.
   - Added input parameters validation for public methods and properties.
   - Removed unnecessary comments.
   - Initialized all fields on declaration where possible.
   - Changed void methods to return values for testing.

3. Classes Refactorings

- **Name Changes:**

    - topResults → topPlayers
    - AddResultInLadder→ AddPlayer
    - ResultQualifiesInLadder→ IsTopResult
    - PrintLadder → to override ToString()

- **Implementation changes**:
    - Added private static field “instance” and public static property “Instance” to implement “Singleton” design pattern.
    - Add method “Clear” for testing.

In class **Player:**

- **Name Changes:**
    - playerName → name

- **Implementation changes**:
    - Added private setters to the properties and verification of the input value.

Struct Message

- **Implementation changes**:
    - Newly created.
    - Moved constants from “UserInputAndOutput” to “Message”.

In class Demo:

- **Implementation changes**:

    - Replaced the endless “while” loop with loop that is finished when user enters command “exit”.

In class **Cell:**

- **Name Changes:**
  - ValueChar → Symbol

- **Implementation changes:**
  - Removed "IsEmpty" method. Replaced with boolean variable in class "Engine".
  - Added private setters and verification to the properties.
  - Extract constants to class "Engine".

Static Class **CosoleIO**

- **Implementation changes:**
  - Replaced all methods with two static methods: "Print" – for printing information on the console and "GetInput" – for reading user input from the console

In class **Direction**

- **Implementation changes:**
  - Created private constructor, public properties with private setters and public static readonly fields for the four directions (New instances of the class can not be created - only the four directinos and their properties can be accessed)

In class **Engine:**

- **Name Changes:**
  - LabyrintStartRow → StartRow
  - LabyrinthStartCol → StartColumn
  - col → column
  - method IsExitPath → HasExit

- **Implementation changes:**
  - Premestvane and ExitPathExists merged -> HasExit
  - Added fields "dummyLabyrith" and "rand", constant "VISITED"
  - Changed the constructor - removed the "rand" parameter
  - Replaced "GetCell" method to "CurrentCell" property

    - Exctracted and changed enum “Direction” to new class
    - Separated method “GenerateLabyrinth” to create and generate the labyrinth in two methods.
    - Changed HasExit implementation and return value- method recursively searches for exit path
    - Merged “TryMove” and “FindNewCellCoordinates” in “TryMove”

In class **Game**

- **Implementation changes**:
    - Introduced fields “scoreboard”, “labyrinth”, “movesCount”, “isRestart” and property “IsExit”
    - Changed constructor to take only one parameter – scoreboard - and verify its value.
    - Exctracted methods: “Play” ,“RunGame”, “AddResultToTopScore”.
    - Removed methods “IsGameOver” and “TryMove”.
    - Separate mehtod “ProcessInput” to ProcessInput”, ProcessMove” and ProcessCommand”

4. Introduced constants:

   In “Scoreboard”

- **NUMBER_TOP_RESULTS = 5**

   In “Engine”

- **EMPTY_ CELL = '-'**
- **WALL_ CELL = 'X'**
- **PLAYER = '*'**
- **VISITED = ‘v’**
- **LABYRITNTH_SIZE = 7**

   In “Message”

- **INVALID_MOVE = "Invalid move!";**
- **WELCOME = "Welcome to \"Labyrinth\" game. Please try to escape." +   "\nUse 'top' to view the top scoreboard, 'restart' to start a new game and \n'exit' to quit the game."**
- **ENTER_MOVE = "Enter your move (L=left, R-right, U=up, D=down): "**
- **ENTER_NAME_FOR_SCOREBOARD =**
- **"Please enter your name for the top scoreboard: "**
- **SCOREBOARD_EMPTY = "The scoreboard is empty."**

- **INVALID_COMMAND = "Invalid command!"**
- **GOOD_BYE = "Goodbye!"**
- **WIN = "Congratulations! You escaped in {0} moves."**

5. Created test classes and DerivedEngine and DerivedGame for testing.

- Created test classes: “CellTest”, “ConsoleIOTest”, “GameTest”, “EngineTest”, “PlayerTest”, “ScoreBoardTest”, “DirectionTest” with 55 test methods covering 86% of the code.
- Created classes “DertivedGame” and “DerivedEngine” which inherit classes “Game” and “Engine” respectively, to test base classes.

6. Public Repository:
   GitHub