Designing Jigsaw Puzzle

→ A jigsaw puzzle is a picture divided into irregularly-shaped pieces that can be assembled together to form a complete picture. Puzzle has fixed number of pieces. They can fit is exact one way

Expectations form the interviewee

→ what does a jigsaw look like. Is there a picture or a pattern on the puzzle?
→ what type of shape does our board have? Is it rectangular, circular or any other shape?
→ what kind of shape do the puzzle pieces have? How many sides does a piece have?
→ How do the piece fit together
→ Will all piece of a jigsaw puzzle be unique.

Requirement Collection

R1: Our board will be in the shape of rectangle

R2: All piece will have four sides that can either have an identation an extrusion, or a flat edge.

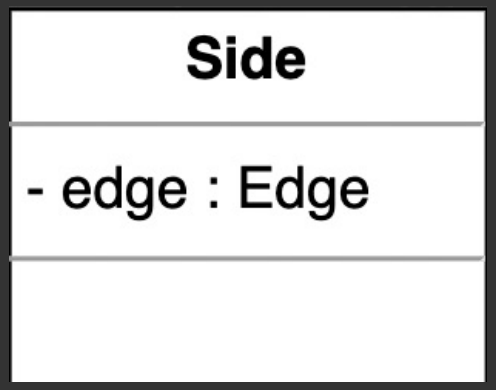
R3: There are four corner pieces, some edges pieces & the remaining ones are the middle pieces. A corner piece has two flat sides, an edge only one flat side, & a middle piece doesn't have any flat edge

R4: All peices will be unique, so only one peice fit with one other price.

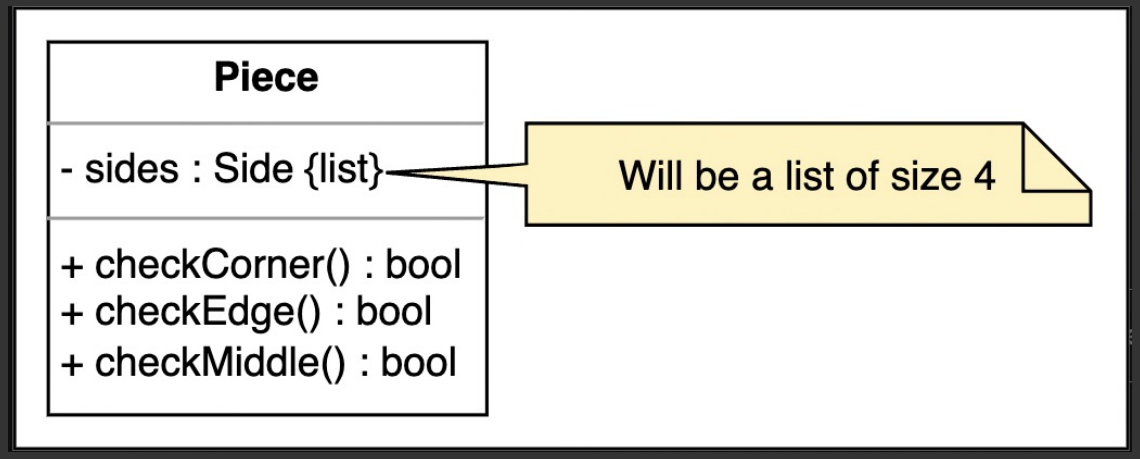
R5: Two peices fit together by the curvature of the identation on one piece matching up to curvature of the extrusion on another.

Class Diagram

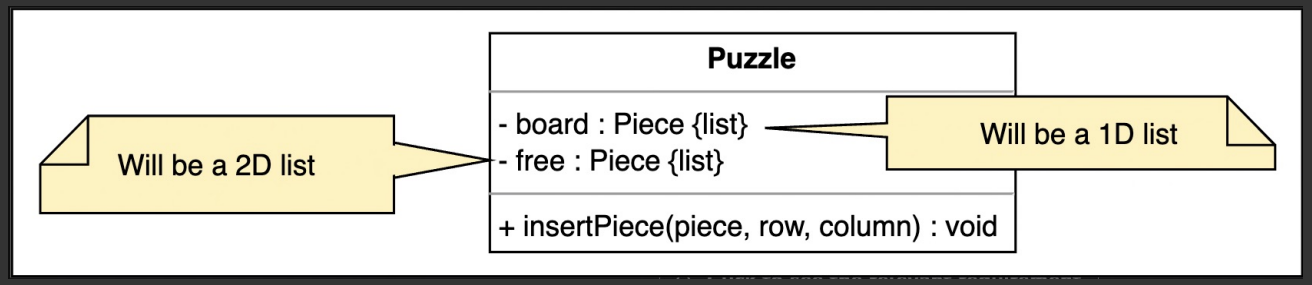
Side



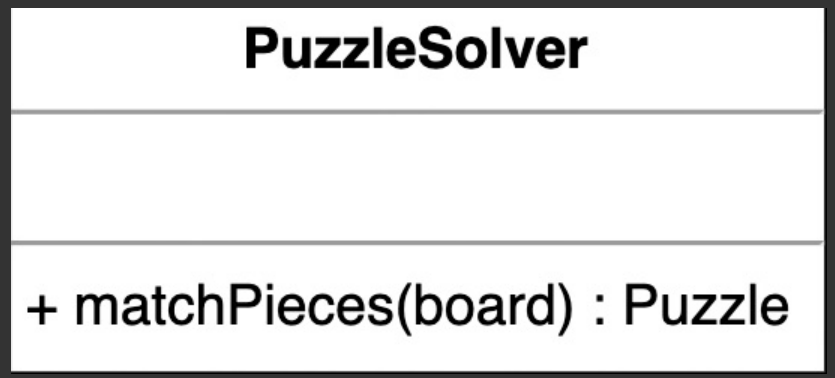
Piece



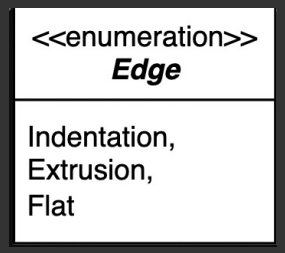
Puzzle



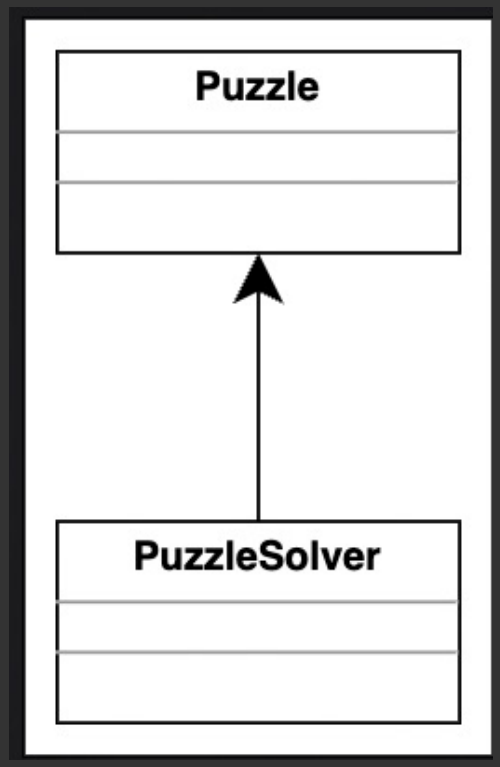
Puzzle Solver



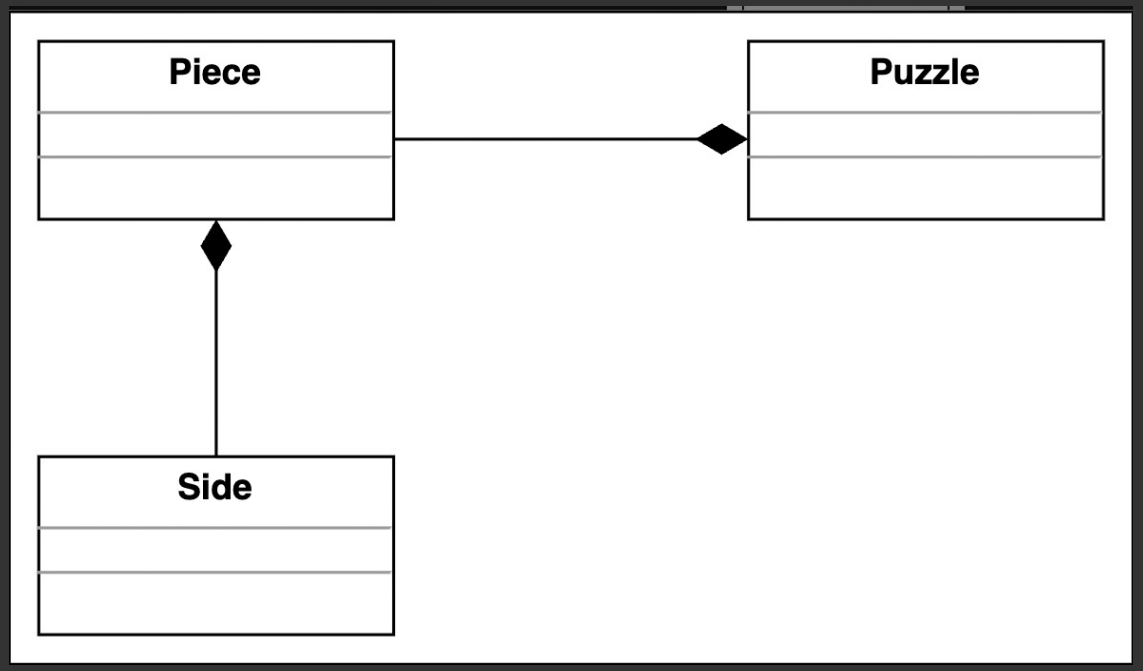
Enumeration



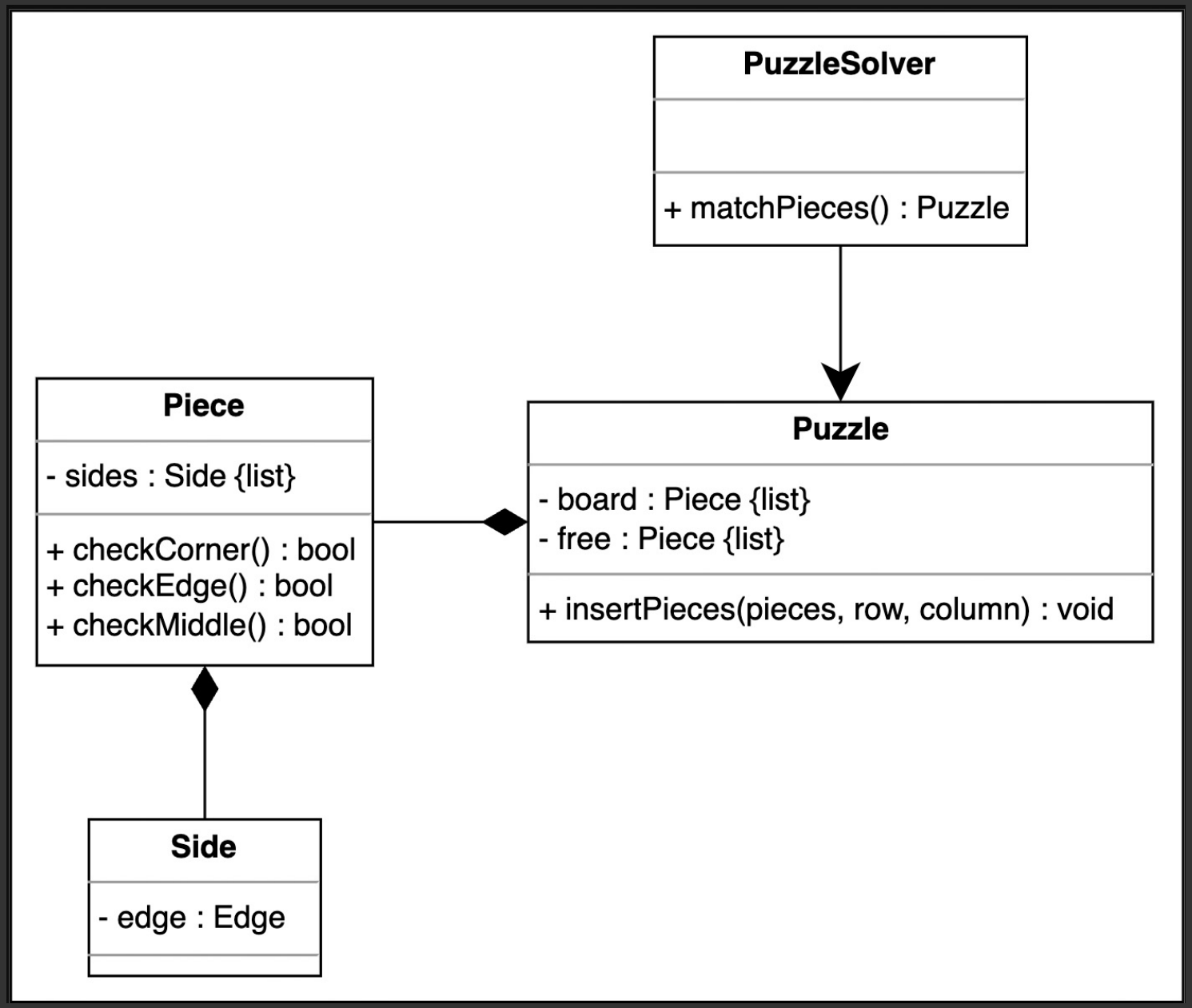
Association



Composition

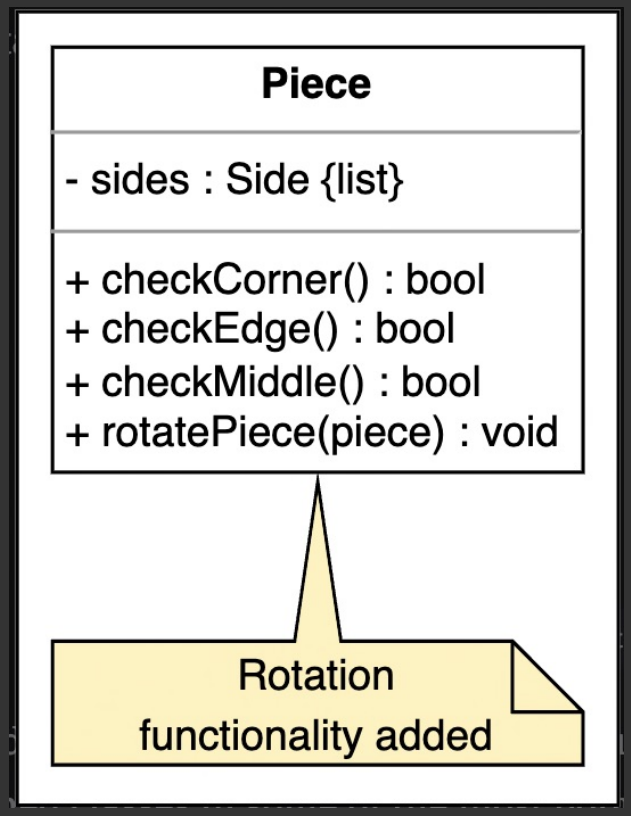


Class Diagram for Jigsaw puzzle



Rotate peice

When addional Requirement of rotate peices.



Code

Enumeration

```
enum Edge {
  INDENTATION,
  EXTRUSION,
  FLAT
}
```

Side

```
public class Side {
  private Edge edge;
}
```

Piece

```
public class Piece {
  private List<Side> sides = Arrays.asList(new Side[4]);
  public boolean checkCorner() {}
  public boolean checkEdge() {}
  public boolean checkMiddle() {}
}
```

Puzzle

```
public class Puzzle {
  private List<List<Piece>> board;
  private List<Piece> free; // represents the currently free pieces (yet to be inserted in board)
  public void insertPiece(Piece piece, int row, int column) {};

  // Puzzle is a singleton class that ensures it will have only one active instance at a time
  private static Puzzle puzzle = null;

  // Created a static method to access the singleton instance of Puzzle class
  public static Puzzle getInstance() {
      if (puzzle == null) {
          puzzle = new Puzzle();
      }
      return puzzle;
  }
}
```

Puzzle Solver

```
public class PuzzleSolver {
  public Puzzle matchPieces(Puzzle board) {}
}
```