# API

Voor ons programma hebben we de volgende API geschreven:

Deel 1, implementeren van een onopgeloste binairo

|  |
| --- |
| Binairo |
| -elements : int[][]  //a table consists of n rows, and n columns  -solvable : Boolean |
| +Binairo(int n)  pre: n > 0  post: has created a Binairo object with n rows and n columns, the value of all elements = -1. Solvable = true.  +set(int row, int col, int val)  pre: 0 <= row < n, 0 <= col < n, val = -1 || val = 0 || val = 1  post: if pre and val = 0 , elements[row][col] = 0, if val = 1 elements[row][col] = 1, else elements[row][col] = -1.  +get(int row, int col) : int  pre: 0 <= row < n, 0 <= col < n  post: if pre returns elements[row][col], returns 7 otherwise  +numRows() : int  post: has returned the number of rows  //since the number of columns equals number of rows, there is no need for a ‘numColumns()’ method.  +toString() : String  post: has returned a String representation of the Binairo object  +solve() : Binairo  post: if the binairo is solvable, return the solution, otherwise, solvable = false, and return the original binairo.  +copy() : Binairo  post: return a copy of the binairo.  +check() : Boolean  post: return false if one of the checkMethods returns “false”, otherwise, return true. |

Deel 2, checkMethods.

Gegeven de specificatie van klasse checkMethods.

|  |
| --- |
| **checkMethods** |
| **-check : boolean** |
| **+checkMethods()**  **post: check = true**  **+noMoreThanTwoRow(Binairo b)**  post: if 3 zeros or ones next to eachother, check = false  **+noMoreThanTwoColumn(Binairo b)**  post: if 3 zeros of ones above eachother, check = false  **+numberOfRows(Binairo b)**  post: if the number of ones, or zeros in the row is more than half of the row length, check = false  **+numberOfColumns(Binairo b)**  post: if the number of ones, or zeros in the column is more than half of the row length, check = false  **+identicalRows(Binairo b)**  Pre: the first two rows are filled with ones and zeros  post: if all the elements of 2 rows match, check = false  **+identicalColumns(Binairo b)**  Pre: all rows are filled with ones and zeros  Post: if all the elements of 2 columns match, check = false  **+getCheck() : boolean**  post: return check |

Deel 3, Gui

Gegeven de specificatie van klasse Gui.

|  |
| --- |
| **Gui** |
| - JTextField[][] binairo;  -JPanel paneel;  -JButton solve;  -JButton zesbijzes;  -JButton achtbijacht;  -JButton tienbijtien;  -JButton veertienbijveertien;  -JLabel foutmelding;  -Container c = getContentPane();  -Font font = new Font("Verdana",Font.PLAIN ,20); |
| **+Gui()**  post: displays a window with a panel with a 2-dimensional textfield array, starting with an empty 6x6 textfield array, and buttons “6x6”, “8x8”, “10x10”, “14x14” and “solve”.  **+input() : Binairo**  post: if the input is a correct value, returns a binairo that contains the values of the input, otherwise, returns a 1x1 binairo.  **+output() : Binairo**  post: displays the binairo in the textfield array.  **+makePanel(int n)**  post: displays an empty nxn textfield array.  **+actionPerformed(ActionEvent e)**  post: if the “solve” button is pressed, and if the input is correct, and the binairo is solvable, the solution is displayed, otherwise, displays an error message. If one of the size buttons is pressed, displays an empty textfield array. |

Deel 4, SolveBinairo

Klasse SolveBinairo bevat alleen een main method.

|  |
| --- |
| **SolveBinairo** |
| //create a new Gui object. |