Cards24GameController

private rootPane:Pane
private findASolutionBtn:Button
private refreshBtn:Button
private verifyBtn:Button
private expressionTextField:TextField
private solutiontextField:TextField
private cardPane: Pane
private time:text
private timer: Timer = new Timer()
private attempts: int = 0
private cardDir:File[] = new File("png").listFiles()
private engine:ScriptEngineManager().getEngineByName("JavaScript")
private possibleNumberOrders:List<Integer[]>new ArrayList<>()
private possibleOperatorOrders:List<string[]> = new ArrayList<>()
private logger = new Logger()

---

public class Cards24GameController()

public void initialize()

void onKey Typed(KeyEvent):void

public void refresh()

private boolean checkEquationNumbersMatch(String)

private void generateNumberOrders(List<Integer>, Stack<Integer>,int)

private void generateOperationOrders(List<String>,Stack<String>,int)

public void shutdown()

1

Cards24MenuController

private rootPane:AnchorPane

private cardDisplay1:ImageView

private cardDisplay2:ImageView

private playButton:Button

---

public class Cards24MenuController

public void initialize()

CardDeck

private static numbersRequiered:List<Integer> = new ArrayList<>()

---

public class CardDeck()

public void addRequiredNumber(Integer)

public static getNumbersRequired():List<Integer>

public static void clearNumbersRequiredList()

Cards24Driver

public static primaryStage:Stage

---

public class Cards24Driver()

public static getPrmaryStage():Stage

public void start(Stage)

public static void main(String[])

Card

private final cardSuite:String

private final cardValue:int

private final cardImage:Image

---

public class Card(String,String)

public getCardSuit():String

public getCardValue():int

public getCardImage():Image

private getNumberFromSuit(String):int

public toString():String

Logger

private file:File

---

public Logger()

public void log(string)