

UML Class Diagrams

Project: ApiArcade

Team: Elon Musketeers

Members: Andy Kawabata, Ryan Cabell, Francisco Serratos Prudencio

Date: 1/1/2020

Tests

TrimSmallWorldsTest

+ main(args : String[]) : void
- buildMap(_input : String[], _expected : String[]) : HashMap<_input : String, _expected : String[]>

factories

GameFactory

+ GAME_ONE : int = 1
+ GAME_TWO : int = 2
+ GAME_THREE : int = 3
+ currentGame : int

+ setCurrentGame(gameNumber : int) : void
+ getCurrentGameInstance

Variables are underlined for static

models

Game

+ gameTitle : String
+ questionText : String
+ imageLabelPairs : ArrayList<HashMap<String,String>>
+ result : String
+ gameOver : boolean = false
+ currentScore : Integer = 0
+ correctAnswer : String
+ currentQuestionNumber : Integer = 1
+ totalQuestions : Integer

+ initialize : void
+ newQuestion : void
+ processAnswer(_givenAnswer : String) : void
+ createMap : Map<String, Object>

Guest

- ISGUEST : boolean

+ login : void
+ save : boolean = false
+ getUuid : String
+ gettid : int

Extends User

Leaderboard

+ game1ID : int = 1
+ game2ID : int = 2

+ getGame1ID : int
+ getGame2ID : int
+ getHighScores(gameID : int, howMany : int) : HashMap<String, Integer>

LoginSession

+ currentUser : User

MovieG

+ NUM_FALSE_MOVIES : int = 4

+ initialize : void
+ swap(_posters : String[], i : int, j : int) : void
+ setNewQuestion : void
+ newQuestion : void
+ processAnswer(_givenAnswer : String) : void
+ resetGame : void
+ shufflePosters(_posterUrls : String[]) : int
+ generateRandomId(_range : int) : int
+ sanitizeDescription(_title : String, _description : String) : String

Extends Game

Sanitizer

+ isArrayEmpty(_array : String[]) : boolean
+ trimSmallWords(_array : String[]) : String[]
+ removePunctuation(_array : String[]) : String[]

Score

+ DEFAULT_HIGH_SCORE : int = -1
+ TABLE : String = "src/storage/scores.csv"
- score : Integer

extends DataObject

+ save : boolean
+ getScoreByUuid(_uuid : String, gameID : int) : int
+ getScoreByUsername(_username : String, gameID : int) : int

TriviaGame

- SCORE_UP : int = 100
- SCORE_DOWN : int = 50

+ initialize : void
+ setGameQuestions : void
+ setNewQuestion : void
+ newQuestion : void
+ processAnswer(_givenAnswer : String) : void
+ resetGame : void
+ checkGameOver : boolean

User

+ TABLE : String = "src/storage/users.csv"
- username : String
- password : String

extends DataObject

+ loadByUsername(_username : String) : User
+ passwordMatches(_password : String) : boolean
+ login : void
+ save : boolean
+ checkErrors(user : User) : String
+ getUsername : String
+ getPassword : String
+ setUsername(_username : String) : void
+ setPassword(_password : String) : void

API

MovieApiInterface

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]

MovieApiAdapter

+ movieApiTranslator : OmdbTranslator

implements MovieApiInterface

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]

OmdbTranslator

- baseUrl : String = "https://api.themoviedb.org/3/movie/"
- apiKey : String = "3ee7fedb5506448980b60224316cdcd0"
- imageBaseUrl : String = "https://image.tmdb.org/t/p/"
- size : String = "w154"

implements MovieApiInterface

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]
+ sanitizeDescription(_title : String, _description : String) : String
+ isArrayEmpty(_array : String[]) : boolean
+ trimSmallWords(_array : String[]) : String[]
+ removePunctuation(_array : String[]) : String[]
+ getResponse(requestUrl : String, movieID : String, similar : boolean) : JSONObject

TriviaApiInterface

+ getGameQuestions : JSONArray
+ getCurrentQuestion(_counter : int) : String
+ getCurrentAnswer(_counter : int) : String

TriviaApiAdapter

+ triviaApiTranslator : OtdbTranslator

implements TriviaApiInterface

+ getGameQuestions : JSONArray
+ getCurrentQuestion(_counter : int) : String
+ getCurrentAnswer(_counter : int) : String

OtdbTranslator

- baseUrl : String = "https://opentdb.com/api.php"
- numOfQuestions : String = "?amount=" + 10
- questionCategory : String = ""; //&category=" + 11
- questionDiff : String = "&difficulty=" + "easy"
- questionType : String = "&type=boolean"
- sessionToken : String = ""
/ urlString : String = baseUrl + numOfQuestions + questionCategory + questionDiff
+ questionType + "&encode=base64" + "&token=" + sessionToken
- retrieveToken : String = "https://opentdb.com/api_token.php?command=request"
- quest : String
- arr : JSONArray

implements TriviaApiInterface

+ apiCall(_urlString : String) : JSONObject
+ getToken : String
+ getGameQuestions : JSONArray
+ getCurrentQuestion : String
+ getCurrentAnswer : String

apiarcade

RunApp

+ primaryStage : Stage
+ mainLayout : AnchorPane

+ main(args : String[]) : void
+ start(primaryStage : Stage) : void
+ showLoginView : void
+ showMainGameView : void
+ showRegisterView : void
+ showLeaderboardsView : void
+ showPassRecoveryView : void
+ showGenericGame : void

App

main(args : String[]) : void

controllers

ForgotPasswordController

implements Initializable

- username : JFXTextField
- newPassword : JFXTextField
- confirmPassword : JFXTextField
- btnResetPassword : JFXButton
- errorLabel : Label

+ initialize(args : String[]) : void
- btnCancel(event : ActionEvent) : void
- btnUpdateInfoAndReturn : void
- _fieldConfPassword(event : KeyEvent) : void
- _fieldPassword(event : KeyEvent) : void
- _fieldUsername(event : KeyEvent) : void
- checkFieldsPopulated : boolean

GenericGameController

- nxtQuestion : Button
- submitBtn : Button
- answerInput : TextField
- _gameTitle : Label
- _questionText : Label
- resultDisplay : Label
- _currentQuestionNumber : Text
- _totalQuestions : Text
- _currentScore : Text
- _imageContainer : HBox

+ initialize(url : URL, rb : ResourceBundle) : void
+ _newQuestion(event : ActionEvent) : void
- _submitAnswer(event : ActionEvent) : void
+ updateView(_data : Map<String, Object>) : void
+ addImagesToView(_imageLabelPairs : ArrayList<HashMap<String,String>>) : void
- _backBtn(event : ActionEvent) : void

LeaderboardController

- game1user1 : Label
- game1user2 : Label
- game1user3 : Label
- game1score1 : Label
- game1score2 : Label
- game1score3 : Label
- game2user1 : Label
- game2user2 : Label
- game2user3 : Label
- game2score1 : Label
- game2score2 : Label
- game2score3 : Label
- userID : Label
- currentGame1Score : Label
- currentGame2Score : Label
- user : User = LoginSession.currentUser
- lb : Leaderboard = new Leaderboard()

+ initialize(url : URL, rb : ResourceBundle) : void
- _mainGamesLink(event : ActionEvent) : void
- _mainGamesPlayNow(event : ActionEvent) : void
- _userLogout(event : ActionEvent) : void

LoginController

- run : RunApp
- _lblXOut : Label
- errorLabel : Label
- logGuest : JFXButton
- username : JFXTextField
- password : JFXPasswordField

+ initialize(url : URL, rb : ResourceBundle) : void
- _btnSignIn(event : ActionEvent) : void
- _hyplnkForgotPassword(event : ActionEvent) : void
- _btnSignUp(event : ActionEvent) : void
- guestLogin(event : ActionEvent) : void

MainGameController

- userID : Label
- highScore : Label
- user : User = LoginSession.currentUser

+ initialize(url : URL, rb : ResourceBundle) : void
_leaderboardLink(event : ActionEvent) : void
- _userLogout(event : ActionEvent) : void
- _gameOneStart(event : ActionEvent) : void
- _gameTwoStart(event : ActionEvent) : void

RegisterController

- _lblErrorMessage : Label
- _lblXOut : Label
- username : JFXTextField
- password : JFXPasswordField
- confirmPassword : JFXPasswordField

+ initialize(url : URL, rb : ResourceBundle) : void
- _btnRegister(event : ActionEvent) : void
- _backBtn(event : ActionEvent) : void

db

DBConnectorInterface

+ createObject(keyValuePairs : Map<String, String>, _table : String) : int
+ readObject(keyValuePairs : Map<String, String>, _table : String) : List<Map<String, String>>
+ updateObject(keyValuePairs : Map<String, String>, _uuid : String, _table : String) : boolean
+ deleteObject(_uuid : String, _table : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>

CSVConnector

implements DBConnectorInterface

+ createObject(_keyValuePairs : Map<String, String>, _table : String) : int
+ readObject(_keyValuePairs : Map<String, String>, _table : String) : List<Map<String, String>>
+ deleteObject(_keyValuePairs : Map<String, String>, _uuid : String) : boolean
+ updateObject(_table : String, _uuid : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>

- getColumnNames(_table : String) : String[]
- buildRow(_columnNames : String[], _keyValuePairs : Map<String, String>) : String[]
- getNextId(table : String) : String
- indexOf(arr : String[], targetString : String) : int
- populateMatchingRows(br : BufferedReader, keyIndex : int, firstValue : String) : ArrayList<String[]>
- alterRow(_columnNames : String[], selectedRow : String[], _keyValuePairs : Map<String, String>) : String[]
- updateFile(csvElements : ArrayList<String>, _table : String) : boolean

DataObject

dataTable : String = ""
id : int = 0
uuid : String

+ createMap : Map<String, String>
+ generateUuid : String
+ getUuid : String
+ gettid : int
+ setUuid(_uuid : String) : void
+ settid(_id : int) : void

DataStoreAdapter

- connector : DBConnectorInterface = new CSVConnector()

+ createObject(map : Map<String, String>, _table : String) : boolean
+ readObject(_keyValue : Map<String, String>, _table : String) : List<Map<String, String>>
+ updateObject(_keyValue : Map<String, String>, _uuid : String, _table : String) : boolean
+ deleteObject(_uuid : String, _table : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>

prototypes

JsonJavaTest

+ getJokeCategoryAndPrint : void

JsonJavaTest2

+ getTwopartJokesAndPrint : void

movieGameAPITest

+ NUMOFQUESTIONS : int = 1
+ DIFFICULTY_EASY : String = "easy"
+ DIFFICULTY_MED : String = "medium"
+ DIFFICULTY_HARD : String = "hard"

+ movieQuestion : String
+ movieDescription : String

HttpClientTest

+ getHttpResponseAndPrint : void

prototypes

JsonJavaTest

+ getJokeCategoryAndPrint : void

JsonJavaTest2

+ getTwopartJokesAndPrint : void

movieGameAPITest

+ NUMOFQUESTIONS : int = 1
+ DIFFICULTY_EASY : String = "easy"
+ DIFFICULTY_MED : String = "meduim"
+ DIFFICULTY_HARD : String = "hard"

+ movieQuestion : String
+ movieDescription : String

HttpClientTest

+ getHttpResponseAndPrint : void

storage

scores.csv

users.csv

<<interface>>
DBConnectorInterface

+ createObject(keyValuePairs : Map<String, String>, _table : String) : int
+ readObject(keyValuePairs : Map<String, String>, _table : String) : List<Map<String, String>>
+ updateObject(keyValuePairs : Map<String, String>, _uuid : String, _table : String) : boolean
+ deleteObject(_uuid : String, _table : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>

CSVConnector

implements
DBConnectorInterface

+ createObject(_keyValuePairs : Map<String, String>, _table : String) : int
+ readObject(_keyValuePairs : Map<String, String>, _table : String) : List<Map<String, String>>
+ deleteObject(_keyValuePairs : Map<String, String>, _uuid : String) : boolean
+ updateObject(_table : String, _uuid : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>

- getColumnNames(_table : String) : String[]
- buildRow(_columnNames : String[], _keyValuePairs : Map<String, String>) : String[]
- getNextId(table : String) : String
- indexOf(arr : String[], targetString : String) : int
- populateMatchingRows(br : BufferedReader,
 keyIndex : int, firstValue : String) : ArrayList<String[]>
- alterRow(_columnNames : String[],
 selectedRow : String[], _keyValuePairs : Map<String, String>) : String[]
- updateFile(csvElements : ArrayList<String>, _table : String) : boolean

DataObject

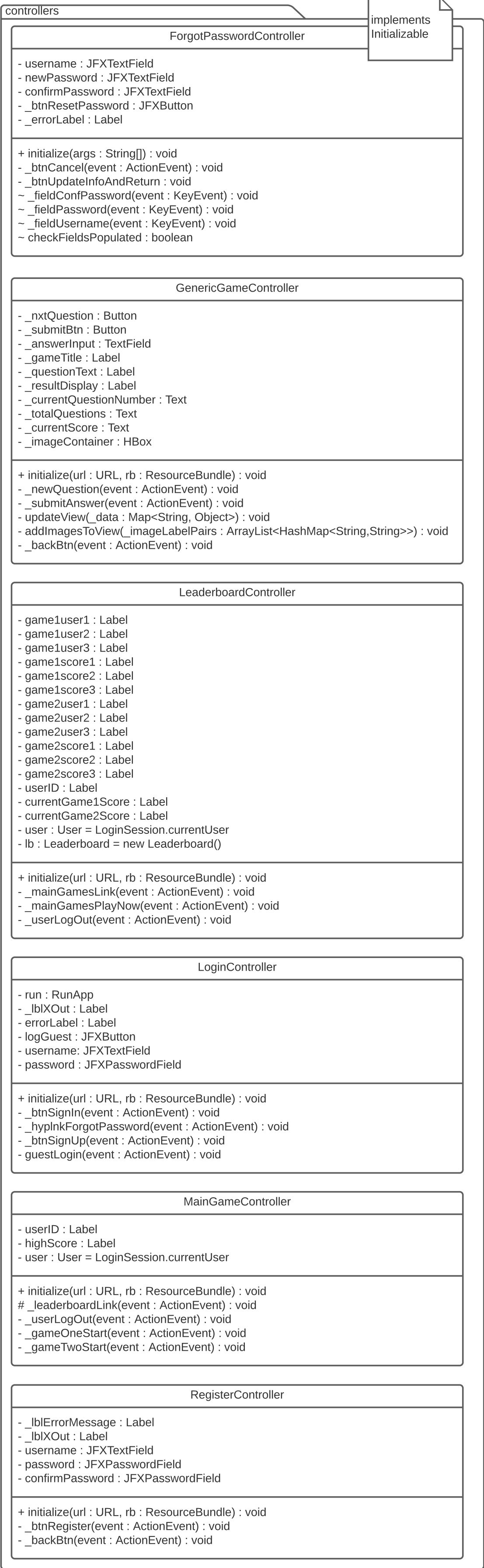
dataTable : String = ""
id : int = 0
uuid : String

+ createMap : Map<String, String>
- generateUuid : String
+ getUuid : String
+ getId : int
- setUuid(_uuid : String) : void
+ setId(_id : int) : void

DataStoreAdapter

- connector : DBConnectorInterface = new CSVConnector()

+ createObject(map : Map<String, String>, _table : String) : boolean
+ readObject(keyValuePair : Map<String, String>, _table : String) : List<Map<String, String>>
+ updateObject(keyValuePair : Map<String, String>, _uuid : String, _table : String) : boolean
+ deleteObject(_uuid : String, _table : String) : boolean
+ getTableRows(_table : String) : ArrayList<String>



RunApp

- primaryStage : Stage
- mainLayout : AnchorPane

+ main(args : String[]) : void
+ start(primaryStage : Stage) : void
+ showLoginView : void
+ showMainGameView : void
+ showRegisterView : void
+ showLeaderboardsView : void
+ showPassRecoveryView : void
+ showGenericGame : void

App

main(args : String[]) : void

**<<interface>>
MovieApiInterface**

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]

MovieApiAdapter

+ movieApiTranslator : OmdbTranslator

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]

implements
MovieApiInterface

OmdbTranslator

~ baseUrl: String = "https://api.themoviedb.org/3/movie/"
~ apiKey: String = "3ee7fedb5506448980b60224316cdcd0"
~ imageBaseUrl: String = "https://image.tmdb.org/t/p/"
~ size : String = "w154"

implements
MovieApiInterface

+ getPosterTitleDescriptionById(id : int) : Map<String, String>
+ getPostersOfSimilarById(id : int, numPosters : int) : String[]
+ sanitizedDescription(_title : String, _description : String) : String
+ isArrayEmpty(_array : String[]) : boolean
+ trimSmallWords(_array : String[]) : String[]
+ removePunctuation(_array : String[]) : String[]
- getResponse(requestUrl : String, movieId : String, similar : boolean) : JSONObject

**<<interface>>
TriviaApiInterface**

+ getGameQuestions : JSONArray
+ getCurrentQuestion(_counter : int) : String
+ getCurrentAnswer(_counter : int) : String

TriviaApiAdapter

+ triviaApiTranslator : OtdbTranslator

+ getGameQuestions : JSONArray
+ getCurrentQuestion(_counter : int) : String
+ getCurrentAnswer(_counter : int) : String

implements
TriviaApiInterface

OtdbTranslator

~ baseUrl : String = "https://opentdb.com/api.php"
~ numOfQuestions : String = "?amount=" + 10
~ questionCategory : String = ""; //&category=" + 11
~ questionDiff : String = "&difficulty=" + "easy"
~ questionType : String = "&type=boolean"
~ sessionToken : String = ""
/ urlString : String = baseUrl + numOfQuestions + questionCategory + questionDiff
+ questionType + "&encode=base64" + "&token=" + sessionToken
~ retrieveToken : String = "https://opentdb.com/api_token.php?command=request"
~ quest : String
~ arr : JSONArray

implements
TriviaApiInterface

+ apiCall(_urlString : String) : JSONObject
+ getToken : String
+ getGameQuestions : JSONArray
+ getCurrentQuestion : String
+ getCurrentAnswer : String

Game

+ gameTitle : String

+ questionText : String

+ imageLabelPairs : ArrayList<HashMap<String,String>>

+ result : String

+ gameOver : boolean = false

+ currentScore : Integer = 0

+ correctAnswer : String

+ currentQuestionNumber : Integer = 1

+ totalQuestions : Integer

Class name is italic for abstract

+ initialize : void

+ newQuestion : void

+ processAnswer(_givenAnswer : String) : void

+ createMap : Map<String, Object>

Guest

- ISGUEST : boolean

Extends User

+ login : void

+ save : boolean = false

+ getUuid : String

+ getId : int

Leaderboard

+ game1ID : int = 1

+ game2ID : int = 2

+ getGame1ID : int

+ getGame2ID : int

+ getHighScores(gameID : int, howMany : int) : HashMap<String, Integer>

LoginSession

+ currentUser : User

MovieGame

+ NUM_FALSE_MOVIES : int = 4

Extends Game

+ initialize : void

+ swap(_posters : String[], i: int, j: int) : void

+ setNewQuestion : void

+ newQuestion : void

+ processAnswer(_givenAnswer : String) : void

+ resetGame : void

- shufflePosters(_posterUrls : String[]) : int

- generateRandomId(_range : int) : int

- sanitizeDescription(_title : String, _description : String) : String

Sanitizer

+ isArrayEmpty(_array : String[]) : boolean

+ trimSmallWords(_array : String[]) : String[]

+ removePunctuation(_array : String[]) : String[]

Score

+ DEFAULT_HIGH_SCORE : int = -1

+ TABLE : String = "src/storage/scores.csv"

- score : Integer

extends DataObject

+ save : boolean

+ getScoreByUuid(_uuid : String, gameId : int) : int

+ getScoreByUsername(_username : String, gameId : int) : int

TriviaGame

- SCORE_UP : int = 100

- SCORE_DOWN : int = 50

+ initialize : void

+ setGameQuestions : void

+ setNewQuestion : void

+ newQuestion : void

+ processAnswer(_givenAnswer : String) : void

+ resetGame : void

- checkGameOver : boolean

User

+ TABLE : String = "src/storage/users.csv"

- username : String

- password : String

extends DataObject

+ loadByUsername(_username : String) : User

+ passwordMatches(_password : String) : boolean

+ login : void

+ save : boolean

+ checkErrors(user : User) : String

+ getUsername : String

+ getPassword : String

+ setUsername(_username : String) : void

+ setPassword(_password : String) : void

factories

GameFactory

- + GAME_ONE : int = 1
- + GAME_TWO : int = 2
- + GAME_THREE : int = 3
- + currentGame : int

Vartiables
are underlined
for static

- + setCurrentGame(gameNumber : int) : void
- + getCurrentGameInstance

Tests

TrimSmallWorldsTest

+ main(args : String[]) : void
- buildMap(_input : String[], _expected : String[]) : Hashmap<_input : String, _expected : String[]>