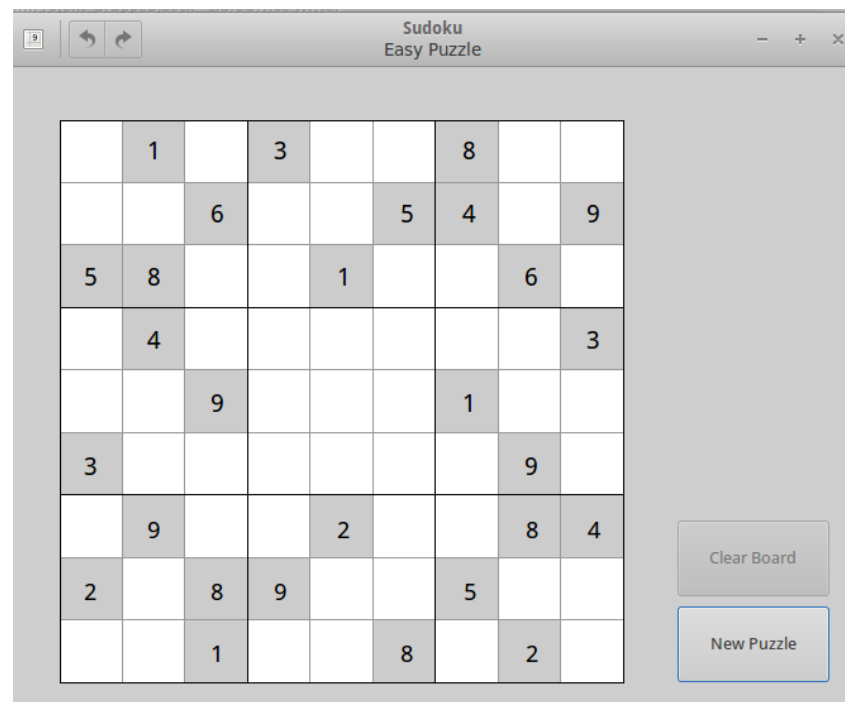


Modul 411 – Sudoku Highscore

Ausgangslage

Sudoku ist ein bekanntes Logik-Rätsel aus Japan. Das Spiel beginnt auf einem 9x9 Feld mit teilweise vorgegebenen Zahlen. Dabei müssen die Zahlen von 1 bis 9 zeilen- und spaltenweise so angeordnet werden, dass in jedem 3x3 Feld ebenfalls die Zahlen von 1 bis 9 enthalten sind. Die Zeit bis zur Fertigstellung wird gemessen. Je schneller der Spieler ist, desto besser.



Die Entwickler der Sudoku-App beauftragen Ihr Team mit der Erstellung einer Highscore-Liste. In der Liste sollen Spielernamen, Datum, das jeweilige Level ("Einfach", "Mittel", "Schwer", "Genie") und die benötigte Zeit gespeichert werden. Für jedes Level werden maximal 10 Einträge gespeichert.

Erforderlicher Funktionsumfang

Implementieren Sie im Team alle notwendigen Klassen und Funktionen innerhalb einer Klasse *HighScoreAdmin*. Einzelne Spielstände werden in einer Klasse *HighScore* abgebildet. Nutzen Sie soweit möglich Datenstrukturen der java.util Collections Bibliothek.

Der Entwicklungsauftrag enthält folgende Schnittstellenbeschreibung für die Klasse *HighScoreAdmin*:

string submit(string name, string dateTime, string level, int finishTime):

gibt Position in entsprechender Liste zurück: "Your rank: 1!" oder "HighScore entries only better than 1354 seconds" wenn die erreichte Zeit grösser als das aktuelle Maximum für das jeweilige Level ist und bereits 10 Einträge existieren.

HighScore[] getHighScores(string level):

liefert alle HighScores des entsprechenden Levels sortiert nach benötigter Zeit in einem Array zurück.

void save():

speichert alle HighScores in einer CSV Datei ab.

void load():

lädt die HighScores aus der gleichen Datei wie save().

Aufgabe

Erstellen Sie die Klasse *HighScoreAdmin* mit allen beschriebenen Methoden. Implementieren Sie in der *main()* Methode einen geeigneten Testrahmen. Darin sollen zunächst einige HighScores geladen werden. Ein kleines Benutzermenü soll die Eingabe neuer HighScores und die Ausgabe je nach Level ermöglichen. Zusätzlich enthält das Benutzermenü eine Ende-Option, womit die aktuellen Highscores gespeichert werden.

Erwartete Ergebnisse:

- Struktogramme für die Methoden *submit(..)* und *getHighScores()*. Nutzen Sie dazu das Programm Structorizer.
- Saubere Implementierung des Programms inklusive inline Dokumentation
- Dokumentation des Projektablaufs inklusive Verantwortlichkeiten
- Kurze Präsentation des Programms und der Struktogramme (max. 10 Minuten)