

# Sudoku

## Dokumentáció

### A programozás alapjai 3

Készítette: Vajda Ádám (AMRREG)



## Architektúra és osztálydiagram

`com.vadam.sudoku.app`

`SudokuApplication` – belépési pont, GUI és szolgáltatások összekötése

`com.vadam.sudoku.config`

`Settings`, `PreferencesManager` – felhasználói beállítások és azok perzisztenciája

`com.vadam.sudoku.controller`

`GameController` – a GUI által hívott vezérlő, amely a szolgáltatásokat koordinálja

`com.vadam.sudoku.exception`

`SudokuException`, `InvalidMoveException`, `PersistenceException`, `PuzzleUnsolvableException` – célzott kivételosztályok

`com.vadam.sudoku.generator`

`SudokuGenerator`, `PuzzleDifficultyEstimator` – megoldott tábla generálása, majd rejtvény generálása egyedi megoldásokkal

`com.vadam.sudoku.model`

`Board`, `CandidateSet`, `Position`, `GameState`, `Move`, `Difficulty`, `GameStatus` – a Sudoku logikai modellje

`com.vadam.sudoku.model.event`

`EventBus, GameEvent, GameEventListener` – eseménykezelő réteg

`com.vadam.sudoku.persistence`

`GameRepository, FileGameRepository, ImportExportService` –  
játékállapot és feladványok tárolása, import/export

`com.vadam.sudoku.service`

`GameService, HintService, TimerService, StatisticsService` – üzleti  
logika, táblaállapot, tippadás, megoldás, időmérés és statisztika

`com.vadam.sudoku.solver, com.vadam.sudoku.solver.strategy`

`Solver, Step, Strategy, SingleCandidateStrategy,`  
`SinglePositionStrategy, NakedPairsStrategy` – Sudoku megoldó  
algoritmus és megoldási stratégiák

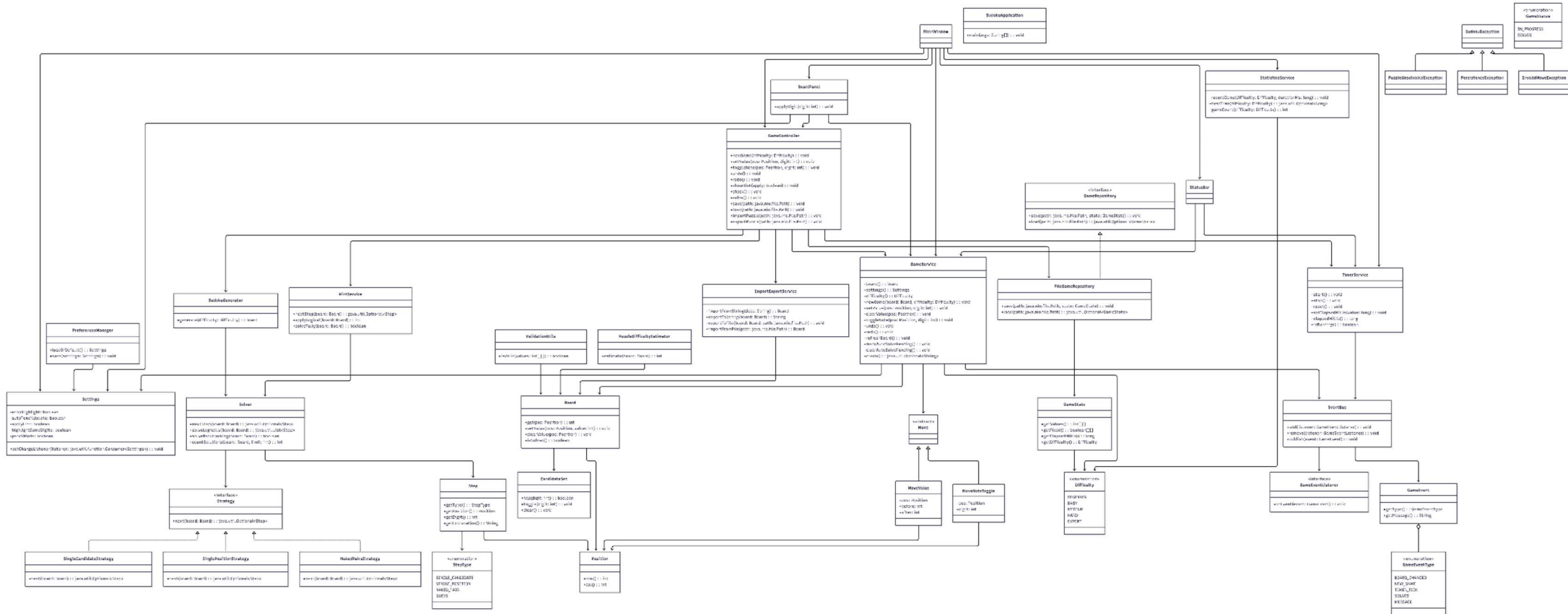
`com.vadam.sudoku.ui`

`MainWindow, BoardPanel, StatusBar` – grafikus felület, menük, toolbar, tábla,  
státuszszor

`com.vadam.sudoku.util`

`ValidationUtils` – Sudoku-szabályok ellenőrzésére szolgáló segédosztály

## Osztálydiagram



## Megvalósítás

A `SudokuApplication` az alkalmazás belépési pontja. A `main` metódus `Swing` EDT-n indítja az alkalmazást, beállítja a rendszer `Look & Feel`-t, létrehozza az `EventBus`-t, betölti a `Settings`-et a `PreferencesManager.loadOrDefault()` segítségével, majd példányosítja a `TimerService`, `GameService`, `StatisticsService`, `SudokuGenerator`, `HintService`, `GameController` és `MainWindow` objektumokat. Indításkor automatikusan egy `EASY` nehézségű játék kezdődik.

A `Settings` olyan opciókat tárol, mint hibakiemelés, ceruzamód, tippek automatikus alkalmazása, stb. A `PreferencesManager`, a `java.util.prefs.Preferences` API segítségével perzisztensen elmenti ezeket felhasználóként, így újraindításkor visszatöltődnek.

A `GameController` a GUI és szolgáltatások közötti vezérlő réteg. Nem tart saját játékállapotot, hanem a `GameService`-re, `SudokuGenerator`-ra, `HintService`-re és `TimerService`-re támaszkodik. Feladata az új játék indítása, tippelés, megoldás, mentés/betöltés és import/export műveletek koordinálása, valamint a hibákról való felhasználói tájékoztatás.

A `GameService` felel az aktuális `Board` állapotért, az `undo/redo` veremért, a nehézségért és a felhasználói beállításokhoz igazodó viselkedésért.

Minden tábla-módosítás után `BOARD_CHANGED` esemény kerül publikálásra az `EventBus`-on, megoldás esetén pedig egyszeri `SOLVED` esemény, amely jelzi, hogy kézi vagy automatikus megoldás volt.

A `Board` egy `9x9`-es táblát reprezentál három mátrixon keresztül: cellaérték (`int[][] values`), fix mezők (`boolean[][] fixed`) és cellánkénti ceruzajegyzetek (`CandidateSet[][] notes`). Fő feladatai a cellaértékek és fix jelöléses kezelése, jelöltek karbantartása, ütközésvizsgálat, érvényesség-ellenőrzés, megoldottság vizsgálata, jelöltek kiszámítása egy cellára, valamint a tábla állapotának be- és kimásolása perzisztenciához.

A `CandidateSet` bitmaszkban tárolja egy cella jelöltjeit (1-9-es bitek). Ez memóriatakarékos és gyors bitműveletekkel teszi lehetővé jelöltek hozzáadását, eltávolítását, kapcsolását, darabszámuk meghatározását, illetve a nyers maszk elérését.

A `GameState` a mentett játékállapot `JSON` reprezentációja: tartalmazza a tábla értékeit, fix mezőit, az eltelt időt és a nehézséget.

A `Move` absztrakt osztály két belső alosztállyal (`ValueMove` és `NoteToggleMove`) írja le az undo/redo szempontjából releváns lépéseket: egy cella értékváltozását, illetve egy jelölt ki- és bekapcsolását.

A Sudoku generátor és solver külön modulban található, hogy a GUI-tól függetlenül tesztelhető és bővíthető legyen.

A Solver logikai stratégiák listájával működik. A `nextStep` egyetlen következő lépést ad (ha van), a `solveLogically` addig alkalmazza a stratégiákat, amíg nem talál több lépést. A `solveBacktracking` visszalépéses keresést végez, mindig a legkevesebb jelölttel bíró cellát választva.

A következő stratégiák állnak rendelkezésre:

- `SingleCandidateStrategy`: olyan cellákat keres, ahol csak egy jelölt lehetséges
- `SinglePositionStrategy`: sor/oszlop/blokk szinten keres számot, amely csak egy helyre illik
- `NakedPairsStrategy`: soronként keresi a naked pair mintákat, és azok alapján eliminál

A `SudokuGenerator` képes egy teljes megoldást készíteni véletlen próbált számokkal és backtrackinggel. A `generate(Difficulty)` a kész megoldásból indul, véletlen sorrendben üríti a cellákat, miközben a `countSolutions` segítségével biztosítja az egyedi megoldást és a célzott mezőszámot.

A perzisztencia modul feladata a játékállapot, statisztikák és feladványok tárolása.

A grafikus felület három komponensből áll:

- `MainWindow`: főablak, eszköztár, menüsor, tábla és státuszsor összeépítése, SOLVED események kezelése, statisztikák megjelenítése
- `BoardPanel`: a Sudoku tábla rajzolása, cellák kijelölése, egér- és billentyűesemények kezelése, ceruzajegyek, hibakiemelés és azonos számok kiemelése
- `StatusBar`: az eltelt idő és rövid állapotüzenetek megjelenítése, folyamatos időfrissítéssel

# Felhasználói kézikönyv

## Futtatás

`mvn exec:java` paranccsal. Tesztelés `mvn test` paranccsal.

## Főablak és kezelőfelület

### Eszköztár (felső sáv)

- *New*: új játék indítása
- *Hint*: tipp kérése
- *Check*: aktuális tábla ellenőrzése
- *Solve*: automatikus megoldás
- *Undo/redo*: visszavonás/újra
- *Save*: játék mentése
- *Load*: játék betöltése

### Tábla

- 9x9-es rács, vastagabb vonalakkal jelölt 3x3-as blokkokkal
- Fix számok félkövén jelennek meg; a kijelölt cella sora, oszlopa és blokkja világosabb háttérszínt kap.
- Hibakiemelés esetén a hibás cellák pirosas háttérrel látszanak. Opcionálisan az összes azonos szám halványan kiemelhető.

### Státuszsor

- Bal oldal: eltelt idő *mm:ss* formátumban.
- Jobb oldal: rövid üzenetek ("Ready", "New Game: EASY", "Solved!")

### Menüsor

- Game menü
  - o *New*: új játék nehézségválasztó párbeszédablakkal
  - o *Import*: feladvány importálása szöveges fájlból
  - o *Export*: aktuális feladvány exportálása szövegfájlba
  - o *Exit*: kilépés
- Options menu
  - o *Error highlighting*: hibakiemelés be- és kikapcsolása
  - o *Auto pencil update*: automatikus ceruzajegy-frissítés be- és kikapcsolása
  - o *Apply hint automatically*: tippek automatikus beírása
  - o *Highlight same digits*: azonos számok halvány kiemelése
  - o *Pencil mode*: globális ceruzajegy mód ki- és bekapcsolása

- Statistics menu
  - o *View*: statisztikák megjelenítése egy táblázatos nézetben (nehézségi szintenkénti játszmák száma és legjobb idő)

## Cellakijelölés

- Egérrel: kattintás a cellára
- Billentyűzettel: nyíl billentyűk mozgatják a kijelölt cellát

## Számbevitel és törlés

- 1–9 beírja a kijelölt cellába a számot (ha nem fix és a ceruzamód ki van kapcsolva)
- 0, Delete, Backspace törli az aktuális cella értékét

## Ceruzajegy mód

- Be- és kikapcsolás: *Options* → *Pencil* mode vagy **P** billentyű
- Ceruzamódban az 1-9 billentyűk a kijelölt cella jelöltjeit állítják át
- Ha az Auto pencil update aktív, egy végleges szám beírása automatikusan kitörli ugyanabban a sorban, oszlopban és blokkban található cellák jelöltjei közül ugyanazt a számot

## Tipp

- Eszköztár: *Hint* gomb, billentyű: **H**
- A program mindig egy szabályos, logikailag indokolható következő lépést ajánl, szöveges magyarázattal. A beállítástól függően automatikusan beírja, vagy csak kiemeli.

## Ellenőrzés

- Eszköztár: *Check* gomb, billentyű: **C**
- Ellenőrzi, hogy van-e szabályt sértő beírás, illetve megoldottnak tekinthető-e már a tábla.

## Automatikus megoldás

- Eszköztár: *Solve* gomb, billentyű: **S**
- Megpróbálja a táblát teljesen megoldani, először logikai lépésekkel, majd szükség esetén visszalépéses stratégiával. Siker esetén az egész tábla kitöltve jelenik meg, az időmérés leáll és a státuszsor „Solved!” üzenetet mutat.

## Undo

- Eszköztár: *Undo* gomb, billentyű: **Z**
- Visszavonja az utolsó műveletet

## Redo

- Eszköztár: *Redo* gomb, billentyű: **Y**
- Újra végrehajtja a legutóbb visszavont műveletet

## Mentés

- Eszköztár: *Save* gomb.
- Fájlválasztóban megadott JSON fájlba menti a teljes játékállapotot (értékek, fix mezők, nehézség, eltelt idő)

## Betöltés

- Eszköztár: *Load* gomb
- Fájlválasztóból kiválasztott mentésből visszaállítja a táblaállapotot és az időmérőt; a ceruzajegyek betöltés után üresek

## Statisztikák

- *Statistics* → *View* menüponttal nyitható meg
- A megjelenő táblázat soronként a nehézségi szinteket (Difficulty), oszloponként a lejátszott játékok számát (Games Played) és a legjobb időt (Best Time) mutatja
- Automatikus megoldással befejezett játékok nem kerülnek bele a statisztikába