

Bedömningsprotokoll, inlämningsuppgift – Sudoku

Namn (programmerare):

Namn (bedömare):

1	<i>Programmets användbarhet</i>	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
1.1	Programmet löser den givna uppgiften enligt anvisningarna. De testfall som hör till uppgiften fungerar.				
1.2	Programmet har ett snyggt och tydligt användargränssnitt. Det innebär t.ex. att användarens uppmärksamhet styrs så att hen gör saker i rätt ordning, att färgval och fonter ger bra kontrast och hög läsbarhet.				
1.3	Programmet är bekvämt att använda.				
1.4	Det är lätt att lära sig använda programmet och att komma ihåg hur man använder det.				
1.5	Programmet reagerar vettigt på felaktiga indata utan att krascha. Det ska vara tydligt vad som blivit fel och vad användaren ska göra åt felet (t.ex. ange ett värde i ett annat intervall).				
1.6	Användaren får feedback i t. ex. en popup-ruta (ej i konsolfönstret) när det behövs. Informationen har en saklig och vänlig ton.				
1.7	Övrigt				

2	Testning av klassen som implementerar SudokuSolver	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
2.1	Det finns JUnit-tester för olika fall: <ul style="list-style-type: none">• tomt sudoku• det lösbara sudokut i uppgiftstexten• olösliga sudokun				
2.2	Alla metoderna från interfacet SudokuSolver anropas minst en gång i något test.				
2.3	De olika testmetoderna har vettiga namn och testar det de är avsedda att testa.				
2.4	Övrigt				
3	Design och implementering	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
3.1	Programmet är uppdelat i modell (Sudoku) och vy (GUI). Modellen känner inte till vyn. (Dvs. det går att kompilera sudoku-klassen/klasserna utan tillgång till GUI-klasserna.)				
3.2	Sudoku: Klassen implementerar interfacet SudokuSolver. Alla metoder från interfacet är implementerade. Indata kontrolleras och exception genereras när det ska enligt dokumentationskommentarerna.				
3.3	Sudoku: Attributen är deklarerade private				
3.4	Sudoku: Den <i>publika</i> metoden för att lösa sudokut ska <i>inte</i> ha parametrar för rad och kolumn eller liknande.				
3.5	Sudoku: Den <i>privata</i> metoden med backtracking-algoritmen har en bra struktur (lätt att följa, inte onödigt lång, inga onödiga loopar etc.).				

3.6	Sudoku: Referensen till den interna matrisen görs inte tillgänglig utanför klassen via returvärde eller parameter. (Kontrollera speciellt konstruktorn, setGrid och getGrid.)				
3.7	GUI: Koden är välstrukturerad.				
3.8	Övrigt				
4	<i>Programmets läsbarhet</i>	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
4.1	Programkoden är lätt att följa och förstå.				
4.2	Uppdelningen i olika metoder är vettig. Metoderna är inte för långa och duplicerad kod undviks.				
4.3	Alla namn är väl valda och återspeglar klassens, metodens, variabelns,... innebörd.				
4.4	Programmet har en vettig indentering och lagom långa rader.				
4.5	Övrigt				