

## 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	<i>Testning av sudokulösarklassen</i>	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
2.1	Alla publika metoder testas				
2.2	Det finns junit-tester för olika fall av sudokun: <ul style="list-style-type: none"> <li>• tomt sudoku</li> <li>• det lösbara sudokut i uppgiftstexten</li> <li>• olösliga sudokun</li> </ul>				
2.3	De olika testmetoderna har vettiga namn och testar det de är avsedda att testa.				
2.4	Övrigt				
3	<i>Design och implementering</i>	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
3.1	Programmet är uppdelat i modell (Sudoku) och vy (GUI). Modellen känner inte till vyn. (Det går att kompilera sudoku-klassen/klasserna utan tillgång till GUI-klasserna)				
3.2	Sudokulösarklassen implementerar interfacet SudokuSolver. Interfacet används som typ i vyn (GUI). (Det är bara när sudoukläsar-objektet skapas som det egna klassnamnet ska användas).				
3.3	Sudoku: Uppdelningen i olika metoder är vettig. Metoderna är inte för långa och duplicerad kod undviks.				
3.4	Sudoku: Attributen är deklarerade private				
3.5	Sudoku: Den <i>privata</i> metoden med backtracking-algoritmen har en bra struktur (lätt att följa, inte onödigt lång.) Lösningar där man med loopar alltid söker från första rutan efter nästa lediga ruta godkänns inte. Istället ska parametrar peka ut aktuell ruta som man utgår ifrån.				

3.6	Sudoku: Referensen till den interna matrisen görs inte tillgänglig utanför klassen. (Se upp med metoder som har en matris som parameter eller returvärde. I så fall bör en kopia skapas.)				
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	Alla namn är väl valda och återspeglar innebörden av klassen, metoden, variabeln,....				
4.3	Programmet har en vettig indentering och lagom långa rader.				
4.4	Övrigt				
5	<i>Dokumentation (Sudoku, ej vyn)</i>	Stämmer	Stämmer delvis	Stämmer ej	Kommentar
5.1	Alla publika metoder i sudokulösar-klassen/klasserna har dokumentationskommentarer som ger den information användaren av klassen behöver.				
5.2	Dokumentationskommentarerna beskriver vad metoden gör och inte hur. Inga implementeringsdetaljer, namn på attribut eller dylikt.				
5.3	I kommentarerna finns HTML-taggar @param, @return och/eller @throws för att beskriva parametrar, returvärde och exceptions.				
5.4	Övrigt				