Course Project
Обектно Ориентирано Програмиране
(OOP with Java)
Play Sudoku

Acceptable Programming Languages:	**Java 1.6.x**
Deadline:	**30th January, 2010*(on the final exam date)***
Instructor	**Dr. Evgeny Krustev**

Настоящият курсов проект е написан от Венета Йосифова, специалност Информатика, 3 курс

Той имаше за задача да се разработи играта Судоку (Sudoku). За тази цел беше използван обектно ориентираният програмен език Java 1.6.0_17 и беше използвана средата за програмиране NetBeans IDE 6.7.1 .

Самият проект е съставен от шест класа SudokuCell.java, SudokuBoardPanel.java, SudokuClient.java, SudokuClientStart.java, SudokuServer.java и SudokuServerStart.java.

SudokuCell наследява JTextField и съдържа описанието на една клетка от таблицата. Тя има две клас данни за реда и стълба в таблицата и също така задава и текстовата стойност на клетката. Има и методи set и get за достъп до данните.

SudokuBoardPanel наследява JPanel. Той има клас данна SudokuCell board[][], която представлява таблицата с клетките. Използван е GridLayout за подреждане на клетките. int countZero; служи за първоначално преброяване на празните клетки, което впоследствие се използва да разпознаване край на играта, кога е решена цялата таблица. Това става с помощта на countValidEnters, която отброява валидните попълнени цифри. Валидацията се извършва от метода boolean validKey(), от вътрешния клас KeyPressHandler. Този клас е KeyAdapter и предефинира метода му void keyTyped(KeyEvent event). В този метод се обработва събитието, настъпило в следствие на въвеждането на цифра и клавиатурата. Ако е била валидна, това се отразява на statusbar-a и тази цифра се поставя в стек Stack undo. Методът void undo() служи за пазене на стойностите въведени от потребителя. Той взима стойностите от стека undo, които са SudokuCell обекти и ги поставя в стека Stack redo; а в таблицата се въвежда празно поле. Методът void undo() има сходно действие, но в таблицата се актуализира стойността на клетката от undo.

SudokuClient е JFrame и имплементира интерфейса Runnable. В него се създава менюто. Зареждането на нова игра става като се обработи събитието в някои от вътрешните класове EasyHandler, MediumHandler, HardHandler. Комуникацията със сървъра се осъществява чрез обектите ObjectInputStream input; ObjectOutputStream output; и методите writeObject и readObject. До сървъра се изпраща съобщение, игра от кое ниво на трудност искаме да се зареди и сървърът ни отговаря със String, съдържащ информация за празните и запълнени клетки. Този низ се подава на метода setBoard, който зарежда стойностите в board[][].

SudokuServer е JFrame и съдържа текстово поле отразяващо свързването на нов клиент. Всеки нов клиент се пуска в нова нишка като се използва Executors.newFixedThreadPool(100); със фиксиран брой до 100. Чрез вътрешния клас

class Player, имплементиращ интерфейса Runnable се осъществява комуникацията със клиента, като му се изпраща игра при заявка от негова страна. Отново се използват обектите ObjectInputStream input; ObjectOutputStream output; и методите writeObject и readObject. Тази операции се намират в безкраен цикъл, за да се даде възможност за многократно зареждане на нова игра.

SudokuClientStart служи за стартиране на клиента. А SudokuServerStart за стартиране на сървъра.

Генерирането на произволни тестове остава да се довърши.

София

Февруари 2010 г.