

Code Review

Gissa Talet

Av Sam Danielsson

```
//Skapar en slumpad siffra från 0 - 100 och returnerar den när man kallar på metoden.
```

INTRODUKTION

I denna uppsats så kommer jag gå igenom Karls logiska fel, onödig kod, om koden har nått upp till kraven av uppgiften, om koden följer Java kod konvention, Karls kommentarer och om projektet är perfekt.

Logiska Fel

Enligt mig så kunde jag inte hitta några logiska fel i Karls arbete och finner att han har gjort ett bra jobb med sin kod i detta fall.

Kraven

Krav

- Använd kontrollstrukturer för att styra programflödet
- Använd någon form av loop
- Använd minst en metod. Metoden ska användas för att evaluera om gissningen är rätt eller fel. Den ska returnera ett värde och ta emot argument. Utöver den får du använda dig av fler metoder om du vill.

Karl har uppnått kraven och jag har sett till att mina ändringar följer uppgiftens krav fortfarande även om det inte var nödvändigt på grund av att jag använder mig av gui.

Onödig kod

Det jag hittade som jag ansåg var onödig kod var att Karl har returnerat Integers istället för en boolean när han kontrollerar om användaren har gissat rätt nummer.

Från:

```
private static int check(int random, int guess) {
    int x = 0;
    if (random == guess)
        x = 3;
    if (random > guess)           //Jämför vä
        x = 2;
    if (guess > random)
        x = 1;
    return (x);
}
```

till :

```
boolean check(int random, int guess){
    setFact(false);

    if(random==guess)
        setFact(true);
    else if (random>guess) {
        Gui.getLog().append("Too low, guess again!\n");
    }

    else
        Gui.getLog().append("too high, guess again!\n");
    return(getFact());
}
```

Java konvention

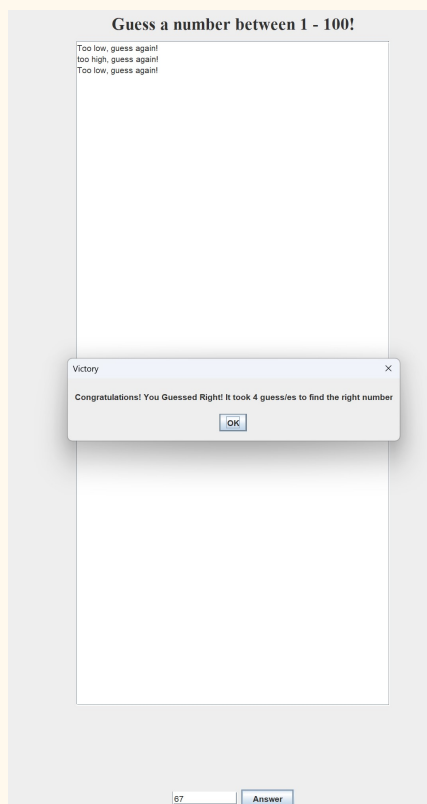
Enligt mig så har Karl följt java konventionen förutom i ett tillfället vilket är när han skapade en metod med stor bokstav.

Kommentarer

Karl har skrivit kommentarer som förklarar tydligt om vad som händer men stämmer inte längre då jag har ändrat projektet från terminal till gui.

Perfekt?

Karls projekt är inte perfekt men han har gjort ett bra jobb med struktur, kommentarer, följa java konventioner, logiska fel och att ha så lite onödig kod som möjligt. De ändringar jag har gjort är mer för att göra användningen av spelet lite smidigare och användbart, lagt till en klass där gui är skapat vilket man kan se i följande bilder.



```

gui() {
    //skapar en ny frame och ser till så den tar upp hela skärmen
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setExtendedState(JFrame.MAXIMIZED_BOTH);
    //Panel för där en knapp samt textfield ska vara
    JPanel bottom = new JPanel();
    setAnswer(new JButton( text: "Answer"));

    //userinput är textfielden, ställer samt in textfields storlek
    setUserInput(new JTextField());
    getUserInput().setColumns(10);
    //En ny label med text samt ställer in font
    JLabel text = new JLabel( text: "Guess a number between 1 - 100!");
    text.setFont(new Font( name: "Serif", Font.BOLD, size: 25));
    //skapar en ny textarea
    setLog(new JTextArea());

    //ställer in textarean's storlek och ser till så att man inte kan ändra på texten i den
    getLog().setColumns(50);
    getLog().setRows(40);
    getLog().setEditable(false);

    //ny Panel som ska sitta centrern av ramen
    JPanel center = new JPanel();
    center.add(new JScrollPane(getLog()), BorderLayout.CENTER);

    // ny panel som ska sitta toppen av ramen
    JPanel top = new JPanel();

    top.add(text);

    //lägger till knappen och textfield i panel bottom
    bottom.add(userInput);
    bottom.add(answer);
    //lägger till allt i ramen
    frame.add(center, BorderLayout.CENTER);
    frame.add(bottom, BorderLayout.SOUTH);
    frame.add(top, BorderLayout.NORTH);
}

```