

```
public class SecretNumber
{
    // Le nombre à deviner
    private int secret = 0;

    // Le nombre d'essais
    private int counter;

    public SecretNumber(int pN)
    {
        secret = (int) (Math.random() * pN) + 1;
        counter = 0;
    }

    public int getCounter()
    {
        return counter;
    }

    public int compareTo(int pGuessedNumber)
    {
        counter++;
        if (pGuessedNumber == secret)
            return 0;
        else if (pGuessedNumber > secret)
            return 1;
        else
            return -1;
    }

    public String guess(int pGuess)
    {
        // pas utilisée dans cet exercice B06...

        String res = "";
        counter++;
        if (pGuess == secret)
        {
            String ending = "th";
            int unit = counter % 10;
            int cent = counter % 100;
            if ((cent != 11) && (cent != 12) && (cent != 13))
            {
                if (unit == 1)
                {
                    ending = "st";
                }
                else if (unit == 2)
                {
                    ending = "nd";
                }
                else if (unit == 3)
                {
                    ending = "rd";
                }
            }
            res = "Well done! You found the secret number at the " + getCounter() + ending + " guess";
        }
        else if (pGuess < secret)
        {
            res = "Your number is too small";
        }
        else
        {
            res = "Your number is too big";
        }
        return res;
    }
}
```

```

public class MainFrame extends javax.swing.JFrame
{
    private SecretNumber secret = new SecretNumber(100);

    public MainFrame()
    {
        initComponents();
    }
    // Skipped: ... initComponents { ... }
    private void nbrTextFieldActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_nbrTextFieldActionPerformed
    { //GEN-HEADEREND:event_nbrTextFieldActionPerformed
        // si le nombre n'a pas encore été deviné
        if (!msgLabel.getText().contains("done"))
        {
            // lire le nombre tapé par l'utilisateur
            int guess = Integer.valueOf(nbrTextField.getText());

            // jouer
            int returnCode = secret.compareTo(guess);

            // afficher un text en fonction du code de retour
            if (returnCode == 0)
                msgLabel.setText("Well done!"); // trouvé!
            else if (returnCode > 0)
                msgLabel.setText("Number too big ...");
            else
                msgLabel.setText("Number too small ...");

            // afficher le nombre de tentatives
            nbrTriesLabel.setText("Number of tries: " + secret.getCounter());
        }
    } //GEN-LAST:event_nbrTextFieldActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[])
    {
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new MainFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel msgLabel;
    private javax.swing.JTextField nbrTextField;
    private javax.swing.JLabel nbrTriesLabel;
    // End of variables declaration//GEN-END:variables
}

```