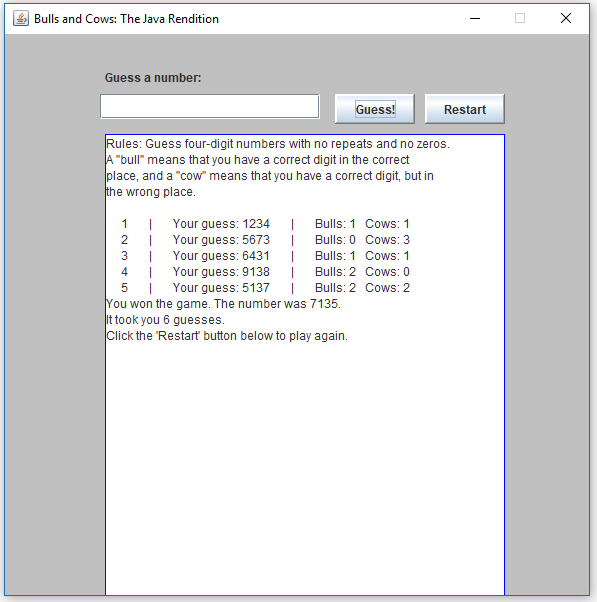
# Bulls and Cows

*A simple Java implementation of the classic Bulls and Cows game. Bulls and Cows (also known as Cows and Bulls or Pigs and Bulls or Bulls and Cleots) is an old code-breaking mind or paper and pencil game for two or more players, predating the commercially marketed board game Mastermind.*

*It is a game that may date back a century or more which uses numbers or words. It is played by two opponents.*



## What are we going to use?

### InteliJ IDEA

Powerful IDE for Java. If you don’t have it, you can download it from here:

<https://www.jetbrains.com/idea/>.

### Java AWT

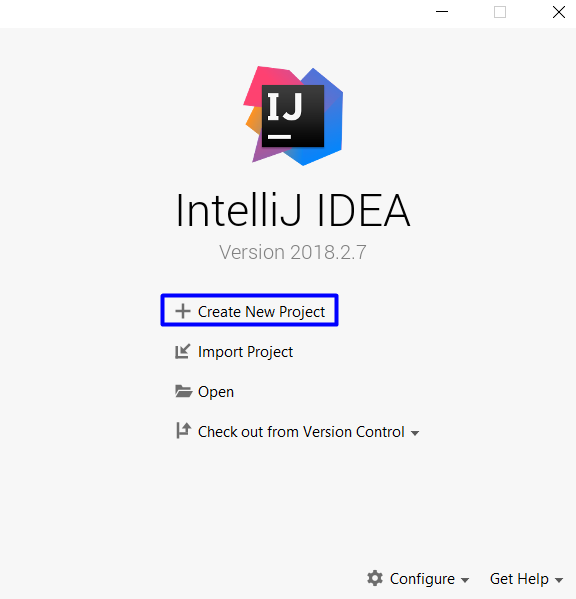
**Java AWT** (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

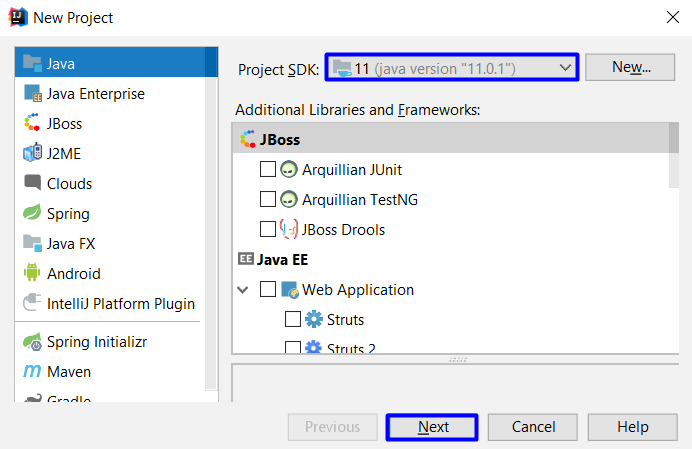
### Java swing

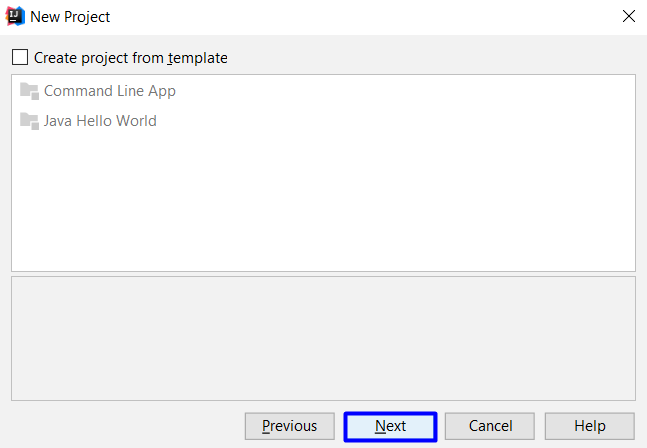
**Swing** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

## Setting up the project

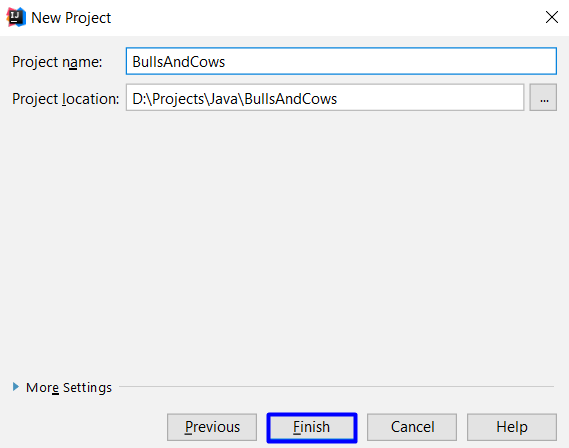
Open your **InteliJ IDEA** create new project.



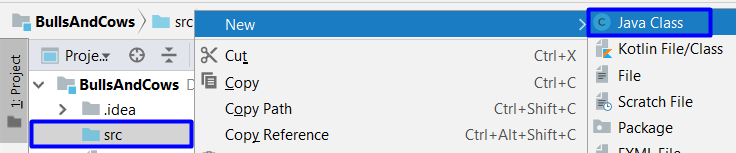


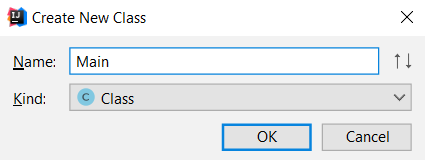


Select directory and name for your project and click Finish.



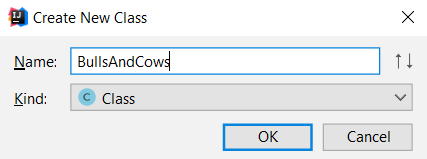
When your project is created you should create a Main class that contains a main method.



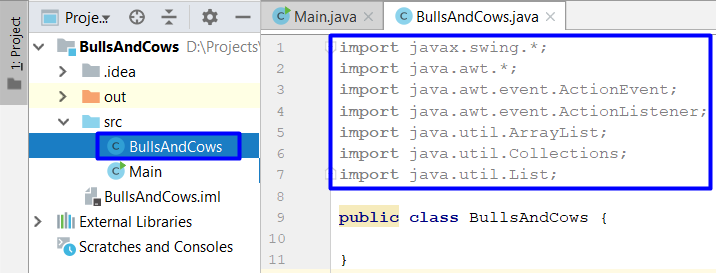


## Create class Bulls and Cows

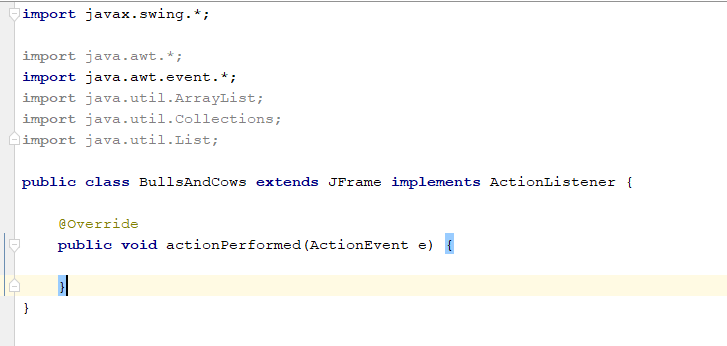
After you create your main method, you need to create another class file in you src directory, that will contain your game logic and implementation, and import some tools that help us create user interface for our application.



Example:



To get some functionality from our helping tools we need to inherit JFrame class and   
implement ActionListener Interface:

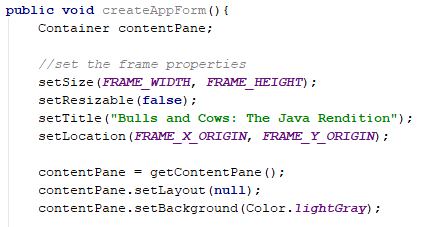


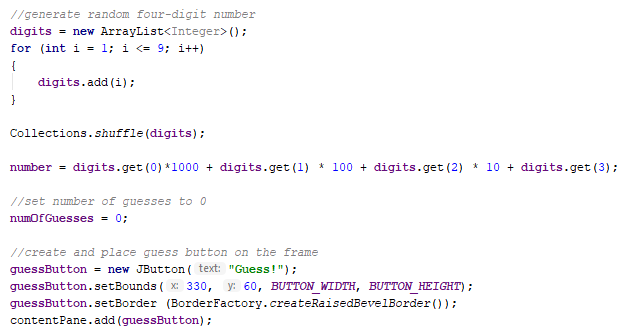
Next you have to create some constants and field that we use later. You can copy form here and paste in the class:

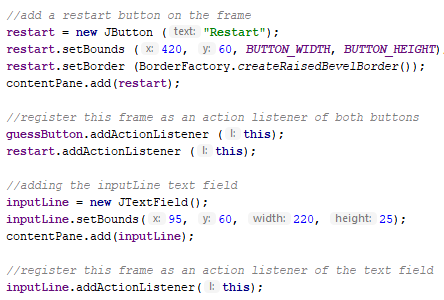
**private static final int *FRAME\_WIDTH*** = 600;  
**private static final int *FRAME\_HEIGHT*** = 600;  
**private static final int *FRAME\_X\_ORIGIN*** = 0;  
**private static final int *FRAME\_Y\_ORIGIN*** = 0;  
**private static final int *BUTTON\_WIDTH*** = 80;  
**private static final int *BUTTON\_HEIGHT*** = 30;  
**private static final** String ***EMPTY\_STRING*** = **""**;  
**private static final** String ***NEWLINE*** = System.*getProperty*(**"line.separator"**);  
**private static final** String ***RULES*** = (**"Rules: Guess four-digit numbers with no repeats and no zeros. "** + **"\nA \"bull\" means that you have a correct digit in the correct \nplace, "** + **"and a \"cow\" means that you have a correct digit, but in \nthe wrong place."** + ***NEWLINE*** + ***NEWLINE***);  
  
**private final** JLabel **instructions** = **new** JLabel (**"Guess a number: "**);  
  
**private** List<Integer> **digits**;  
  
**private** JButton **guessButton**;  
  
**private** JButton **restart**;  
  
**private** JTextField **inputLine**;  
  
**private** JTextArea **textArea**;  
  
**private int number**;  
  
**private int numOfGuesses**;

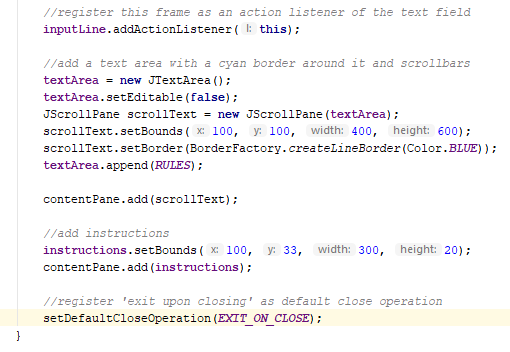
Just below our constants and fields, we define a method that sets up our app properties to the form. It should contain the following logic:

* set the frame properties
* generate random four-digit number
* set number of guesses to 0
* create and place guess button on the frame
* add a restart button on the frame
* register this frame as an action listener of both buttons
* adding the inputLine text field
* register this frame as an action listener of the text field
* add a text area with a cyan border around it and scrollbars
* add instructions
* register 'exit upon closing' as default close operation:

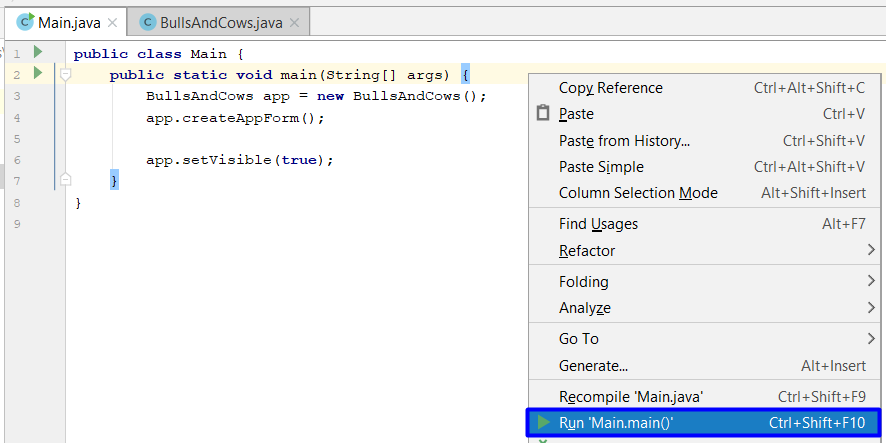




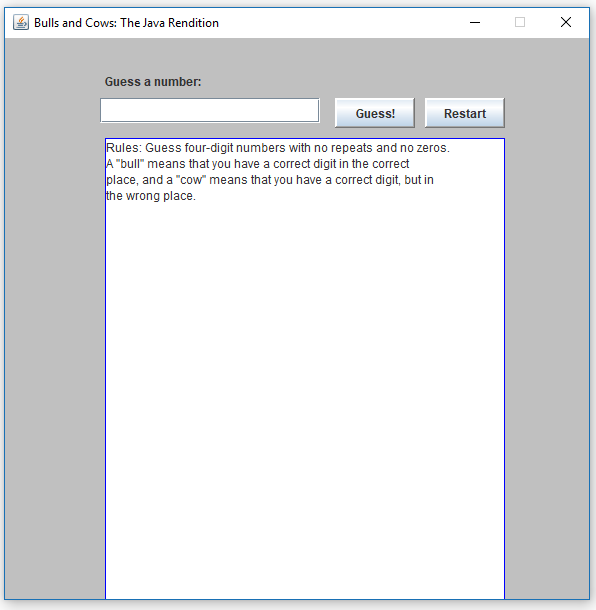




Let's now go to our Main method and see what we have done so far. Open your class Main and write the following code inside and run your application.

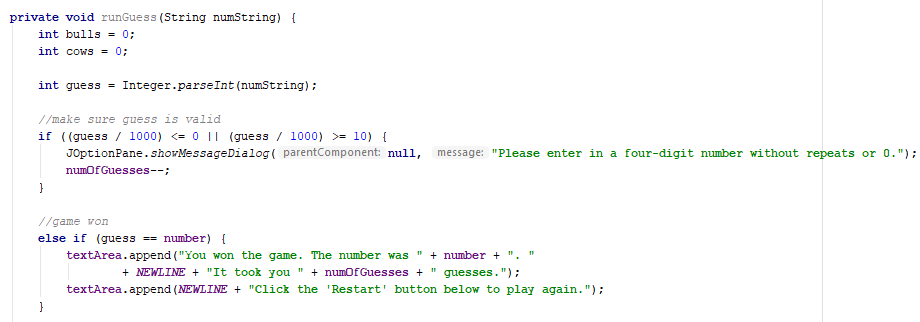


If everything is ok your app should look like this:

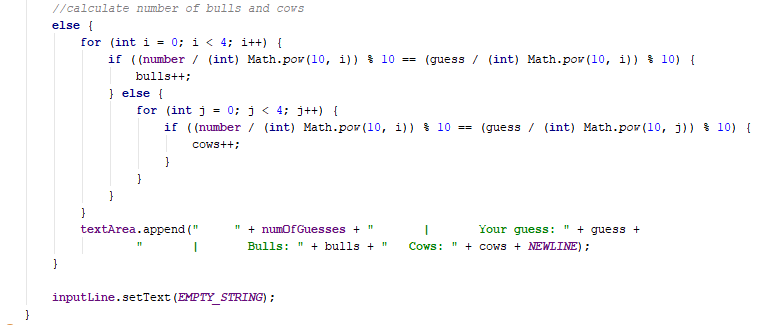


After our form is done, it`s time to give it some functionality. Let's start with our input. After the user has written a number in the input field we should validate it, check how many cows and bulls the user has, and return a result. Let`s create a method called runGuess(String imput) in the BullsAndCows.java class. This method will do the following:

* make sure the guess is valid



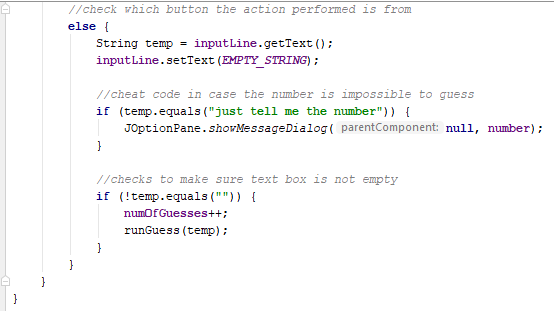
* calculate the number of bulls and cows:



## Method Action Performed implementation

Looks like we are ready with the functionality, but our app does not know how to use it. To resolve this problem we need to write some other logic to our actionPerformed()method, which we created earlier.





## Run your app

If everything is ok, just run your application from the main method. 😊 Enjoy!

