

**PRT452 SOFTWARE ENGINEERING: PROCESS AND TOOLS**

**ASSIGNMENT- 1**

**Submitted by**

**ASHISH KUMAR REDDY LANKALA (s321330)**

**Master of Information Technology (Software Engineering)**

**DOCUMENT ON REFACTORING**

**NORMAL CODE**

Here is the normal code written in java to guess a number in between 1 and 100 and to return the number of trials made to guess the random number generated

**import** java.util.Scanner;

**import** java.util.Random;

**public** **class** GuessingGame {

**public** **static** **void** main(String[] args ) {

Random rand=**new** Random();

**int** secretNumber = rand.nextInt(100);

**int** tries = 0;

System.***out***.println("Secret number is " + secretNumber);

Scanner keyboard = **new** Scanner(System.***in***);

**int** guess = 0;

**do** {

System.***out***.print("Enter a guess (1-100): or q to quit ");

String quit =keyboard.nextLine();

**if**(quit.equalsIgnoreCase("q")) {

System.***out***.println("thank you");

**break**;

}

**else**

tries++;

guess = keyboard.nextInt();

**if** (guess == secretNumber)

{

System.***out***.println("Your guess is correct.");

System.***out***.println("It took you "+ tries + " tries.");

**break**;

}

}

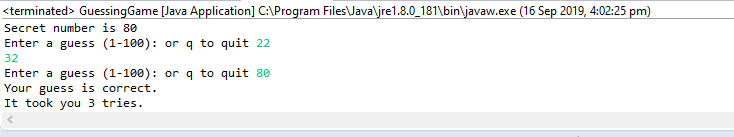
**while** (**true**);

keyboard.close();

}

}

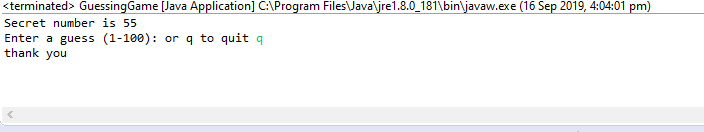
Below I am including the code and screenshots of code and the result obtained



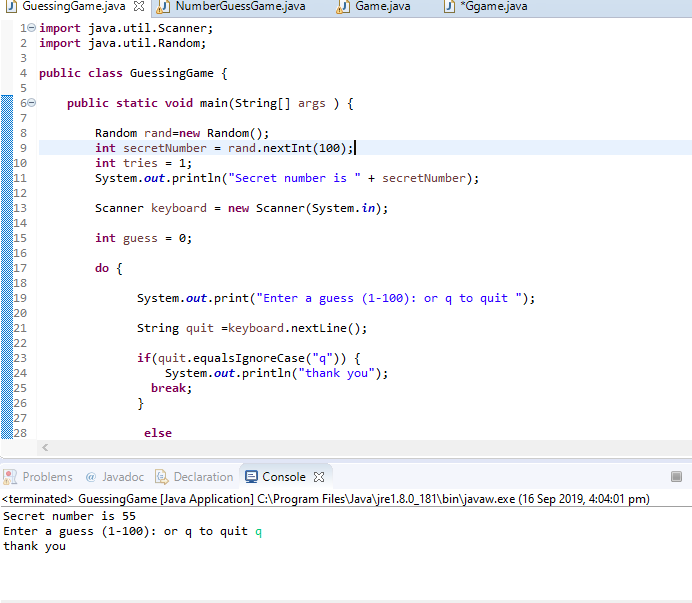
In the above screenshot you can see the result obtained the code shows the number of trails attempted to guess the random number

Note: I have printed the guess number which is randomly generated to find the answer easily

It can be removed if not required



In the above screenshot you can see that the program has terminated when entered q



This image shows the whole code and result in the IDE

I would like to refactor that code so that I do not duplicate Statements. I always need to check condition1 before any other condition.

**REFACTORED CODE**

Refactoring is changing the structure of the code without changing its external behaviour

There can be some code smells recognized in the normal code which shows the bad behaviour of the code

The code smells can be vanished by using the method of refactoring where the functionality of the code does not changes rather it makes the code more understandable and clear

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** Ggame {

**public** **static** **void** main(String[] args) {

Random r = **new** Random();

**int** tries = 0;

**int** secretNumber = r.nextInt((100 - 1) + 1) + 1;

System.***out***.println("Secret number is " + secretNumber);

Scanner input = **new** Scanner(System.***in***);

String number = "";

**boolean** check = **false**;

**do** {

System.***out***.print("Enter Guess Number between 1 -100 or press q to quit ");

**try** {

number = input.next();

**if**(number.equals("q"))

{

check = **false**;

}

**else**

{

**if**(secretNumber == Integer.*parseInt*(number))

check = **false**;

**else**

check = **true**;

tries = tries + 1;

}

}

**catch**(Exception ex)

{

System.***out***.println("Error: "+ex.getMessage());

}

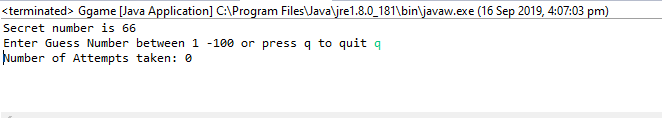
}**while**(check);

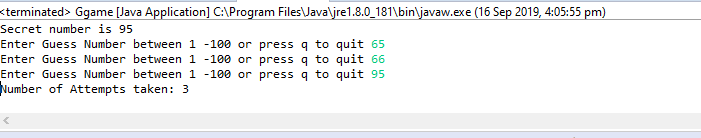
input.close();

System.***out***.println("Number of Attempts taken: "+tries);

}

}



Results for the refactoring code

**GITHUB PATHLINK**

https://github.com/AshishKumarReddyL/PRT452.git