

Forum:[Hack Forums](#)

Thread: [Java OOP](#)

Consulting Detective - July 13, 2018, 3:59 am

Hi Guys,

I am learning Java right now and I'm currently on the topic of OOP basics. I wrote this small program, just wondering what you guys think.

How this program works is that it creates 3 players, generates 3 mystery numbers and generates 3 number guesses for the players. It then compares the numbers and check if the players guessed right. The console shows which player(s) won.

I have not learned how use the input stream yet so no hate :). But I'm sure it's not that difficult since I have programmed in C before.

Right now I'm trying to focus on learning OOP as this is the first time I'm tackling it.

I made 3 classes. First is the game launcher, where my objects are created and methods are called. Second consists of a method that generate a random number between 1-9 for each of the three players and another method to compare the mystery number and the player guess number. Lastly, a class for generating user guesses randomly.

PHP Code

```
<?php
public class GameLauncher{
    public static void main(String args[]){
        GuessGame p1 = new GuessGame();
        GuessGame p2 = new GuessGame();
        GuessGame p3 = new GuessGame();

        p1.mysteryNum = p1.genRandomNum();
        p2.mysteryNum = p2.genRandomNum();
        p3.mysteryNum = p2.genRandomNum();

        Player g1 = new Player();
        Player g2 = new Player();
        Player g3 = new Player();

        g1.playerGuess = g1.genPlayerNum();
        g2.playerGuess = g2.genPlayerNum();
        g3.playerGuess = g3.genPlayerNum();

        p1.compareNums(p1.mysteryNum, g1.playerGuess, 1);
        p2.compareNums(p2.mysteryNum, g2.playerGuess, 2);
        p3.compareNums(p3.mysteryNum, g3.playerGuess, 3);

    }
}
```

PHP Code

```

<?php
class GuessGame{
    int mysteryNum;
    int genRandomNum(){
        return((int)Math.floor(Math.random() * 10));
    }

    void compareNums(int num1, int num2, int playerNum){
        System.out.print("P"+playerNum+": Mystery #="+num1+" Guess="+num2);
        if(num1 == num2){
            System.out.println(" ---> WIN");
        }
        else{
            System.out.println(" ---> LOST");
        }
    }
}

```

PHP Code

```

<?php
class Player{
    int playerGuess;
    int genPlayerNum(){
        return((int)Math.floor(Math.random() * 10));
    }
}

```

Tenno - July 13, 2018, 8:29 am

If you want to accept user input, you can use the Scanner utility. Simply import java.util.Scanner and create a new instance of the scanner like this:

PHP Code

```

<?php
// Import Scanner class.
import java.util.Scanner;

public class Main{

    public static void main(String args[]){
        // Create new instance of the scanner. System.in means the system inputstream.
        Scanner sc = new Scanner (System.in)

```

```

// Get integer integer input and store it
int i = sc.nextInt();

```

```

//Get string input and store it
String s = sc.nextLine();
// You can also use scanner.next()., but with next() it only reads the input until a space. nextLine() reads the whole line.
}
}

```

You can read more about the scanner here:

<https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

I would also recommend that you look into getters and setters. In your program, you have created a few fields (class variables) that are public to every class within your package. By the looks of it, you haven't looked at access modifiers yet and these are crucial to the integrity of your program.

In some of your classes, you define a variable like this:

PHP Code

```
<?php
```

```
int playerGuess;
```

By not giving this an access modifier (f.ex private), any class that is within your package can access this variable and change its value. This makes your program vulnerable to unwanted changes by human error or sabotage. This is where we would use getters and setters. These methods allows us to access and use a field without directly changing the value which could affect other parts of your program. A getter is simply a method that returns the value of something, for example the playerGuess field. It looks like this:

PHP Code

```
<?php
```

```
private int playerGuess;
```

```
public int getPlayerGuess(){
```

```
return this.playerGuess.
```

```
}
```

This method is public, meaning any class can use this method, and its returntype is an integer. "this" just makes sure that it returns the correct playerGuess field.

A setter is a method that allows us to safely change the value of a variable. A setter looks like this:

PHP Code

```
<?php
```

```
private int setPlayerClass;
```

```
public void setPlayerGuess(int guess){
```

```
this.playerGuess = guess;
```

```
}
```

This takes an integer as the parameter and changes the playerGuess field value.

We use these methods so we dont need to directly access the playerGuess field, and we can then declare the playerGuess field as private. This is a part of encapsulation, which is a fundamental part of OOP.

You should do your own research on everything I mentioned, as I only scratched the surface. If you have any more questions, just PM me :) I can also rewrite your program the way I would do it if you want ?

Consulting Detective - July 13, 2018, 5:27 pm

Wow Tenno. This is overwhelming. But thank you nonetheless. I am actually reading the "Head First Java" book by O'Reilly from cover the cover and using it as a study guide. But if I am stuck with something I will definitely come back here.

EDIT: I just learned about encapsulation and its benefits. Thank you.

GoodVibes - July 14, 2018, 4:28 pm

One thing I've learned from my programming experiences with Demmonic and GenericSkid is that you should always call the Garbage Collector. If you don't in bigger programs, you'll keep making these variables you're making and eventually you're run out of memory, and throw the OutOfMemoryException. To avoid that, you can just make a try catch where you run out of memory and continue the program code in the exception block, or you can just call the Garbage Collector which is the proper way to go. Thanks, Tenno.

GenericSkid - July 15, 2018, 1:22 am

(Jul 14, 2018 04:28 PM)GoodVibes Wrote: [\[CITE\]](#) One thing I've learned from my programming experiences with Demmonic and GenericSkid is that you should always call the Garbage Collector. If you don't in bigger programs, you'll keep making these variables you're making and eventually you're run out of memory, and throw the OutOfMemoryException. To avoid that, you can just make a try catch where you run out of memory and continue the program code in the exception block, or you can just call the Garbage Collector which is the proper way to go. Thanks, Tenno. - Literally everything I post is garbage
?

Caliban - July 16, 2018, 12:55 pm

I recommend you read the java docs and you also learn about design patterns because right now you have a bad case of spaghetti code.

If you want a secret tip, look up the Java learning path on pluralsight, its very high quality and you can also learn about more advanced topics like concurrency, Spring, JDBC, Generics etc.