Intro2CS HW 02

יואב שרת

12-31-2023

1. Divisors

```
public class Divisors {
    public static void main (String[] args) {
        int number_from_usr = Integer.parseInt(args[0]);

        for (int i = 1; i<=number_from_usr; i++){
            if(number_from_usr % i == 0){
                System.out.println(i);
            }
        }
    }
}
```

2. Reverse a string

```java
public class Reverse {
    public static void main (String[] args){
        String usr_input = args[0];
        int input_length = usr_input.length();

        // prints reversed usr input
        for(int i = input_length-1; i>=0; i--){
            System.out.print(usr_input.charAt(i));
        }
        System.out.println();

        //finding the middle char, and outputs it
        int middle_index = input_length % 2 == 0 ? (input_length/2)-1 :
                        ((input_length+1)/2)-1;
        System.out.println("The middle character is " +
                            usr_input.charAt(middle_index));
    }
}
```

3. Lucky streak

```
public class InOrder {
    public static void main (String[] args) {
        int MAX_RAND_VAL = 10; // range is [0, 10)
        int new_rand = (int)(Math.random() * MAX_RAND_VAL);
        int last_rand_val = -1;
        do {
            last_rand_val = new_rand;
            System.out.print(last_rand_val + " ");
            // generate new random number
            new_rand = (int)(Math.random() * MAX_RAND_VAL);
        } while (new_rand >= last_rand_val);
    }
}
```

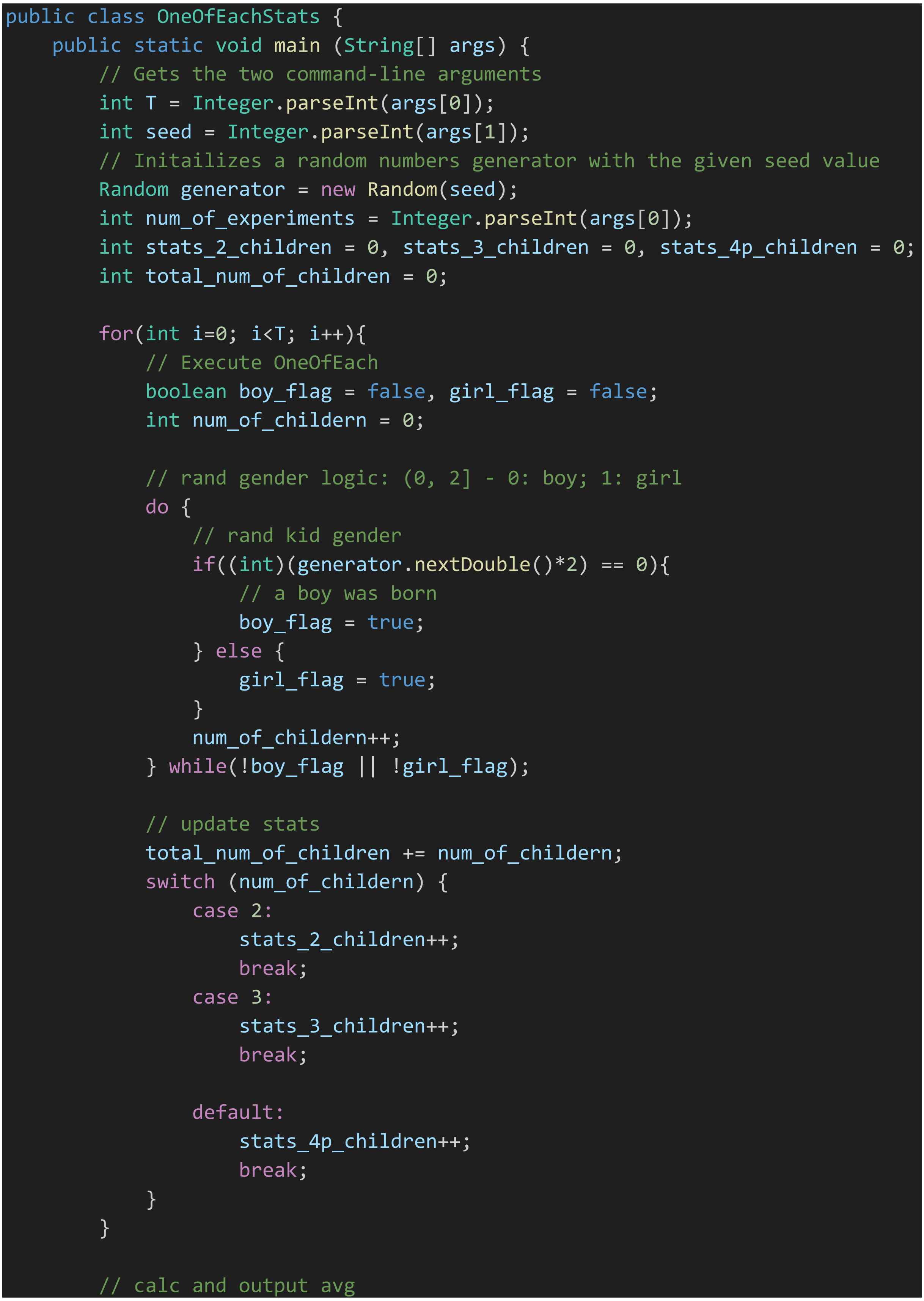
4. Perfect Numbers

```java
public class Perfect {
    public static void main (String[] args) {
        int usr_input = Integer.parseInt(args[0]);
        int sum_devisors = 0;
        String output = usr_input + " is a perfect number since " + usr_input + " = ";

        for(int i=1; i<usr_input; i++){
            if(usr_input % i == 0){
                sum_devisors += i;
                output += (i + " + ");
            }
        }
        output = output.substring(0, output.length()-3);  //removing the last " + "

        if(sum_devisors != usr_input){
            //User input is a perfect number
            output = usr_input + " is not a perfect number";
        }

        System.out.println(output);
    }
}
```

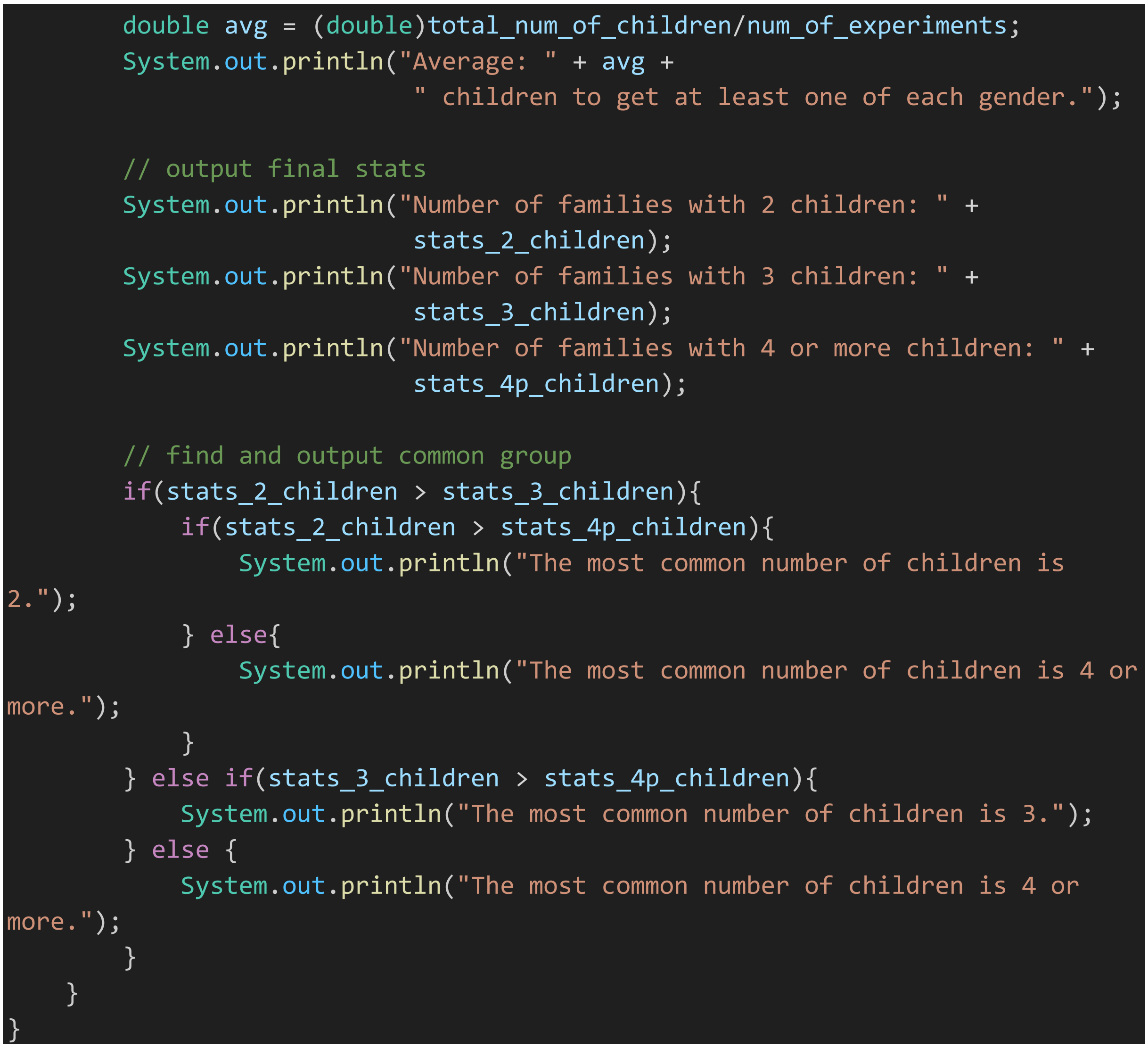
5. Damka Board

```java
public class DamkaBoard {
    public static void main(String[] args) {
        int input_board_dimensions = Integer.parseInt(args[0]);

        for(int i = 0; i<input_board_dimensions; i++){
            String output_token = (i % 2 == 0) ? "* ": " *";
            for(int j = 0; j<input_board_dimensions; j++){
                System.out.print(output_token);
            }
            System.out.println();
        }
    }
}
```

6. One of Each Stats



```java
public class OneOfEachStats {
    public static void main (String[] args) {
        // Gets the two command-line arguments
        int T = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);
        // Initailizes a random numbers generator with the given seed value
        Random generator = new Random(seed);
        int num_of_experiments = Integer.parseInt(args[0]);
        int stats_2_children = 0, stats_3_children = 0, stats_4p_children = 0;
        int total_num_of_children = 0;

        for(int i=0; i<T; i++){
            // Execute OneOfEach
            boolean boy_flag = false, girl_flag = false;
            int num_of_childern = 0;

            // rand gender logic: (0, 2] - 0: boy; 1: girl
            do {
                // rand kid gender
                if((int)(generator.nextDouble()*2) == 0){
                    // a boy was born
                    boy_flag = true;
                } else {
                    girl_flag = true;
                }
                num_of_childern++;
            } while(!boy_flag || !girl_flag);

            // update stats
            total_num_of_children += num_of_childern;
            switch (num_of_childern) {
                case 2:
                    stats_2_children++;
                    break;
                case 3:
                    stats_3_children++;
                    break;

                default:
                    stats_4p_children++;
                    break;
            }
        }

        // calc and output avg
```



```
        double avg = (double)total_num_of_children/num_of_experiments;
        System.out.println("Average: " + avg +
                            " children to get at least one of each gender.");

        // output final stats
        System.out.println("Number of families with 2 children: " +
                            stats_2_children);
        System.out.println("Number of families with 3 children: " +
                            stats_3_children);
        System.out.println("Number of families with 4 or more children: " +
                            stats_4p_children);

        // find and output common group
        if(stats_2_children > stats_3_children){
            if(stats_2_children > stats_4p_children){
                System.out.println("The most common number of children is
2.");
            } else{
                System.out.println("The most common number of children is 4 or
more.");
            }
        } else if(stats_3_children > stats_4p_children){
            System.out.println("The most common number of children is 3.");
        } else {
            System.out.println("The most common number of children is 4 or
more.");
        }
    }
}
```