Divisors

```java
/**
 *  Gets a command-line argument (int), and prints all the
divisors of the given number.
 */
public class Divisors {
    public static void main (String[] args) {
        // Parse the command-line argument as an integer
        int number = Integer.parseInt(args[0]);

        // Iterate through numbers from 1 to the given number
and prints the divisors
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                System.out.println(i);
            }
        }
    }
}
```

Reverse

```
/**
 * Prints a given string, backward. Then prints the middle
character in the string.
 * The program expects to get one command-line argument: A
string.
 */
public class Reverse {
    public static void main(String[] args) throws Exception {

        // Get the input string from the command-line argument
and declare some args
        String word = args[0];
        String reverseWord = "";
        int middleIndex = 0;

        // Calculate the middle index of the string
        middleIndex = (word.length()%2 == 0) ? (word.length()/2)
: (word.length()/2) + 1;

        // Reverse the string
        for (int i = word.length() - 1; i >= 0; i--) {
            reverseWord = reverseWord + word.charAt(i);
        }

        // Print the reversed string and the middle char of the
string
        System.out.println(reverseWord);
        System.out.println("The middle character is " +
word.charAt(middleIndex - 1));
    }
}
```

InOrder

```java
/**
 *  Generates and prints random integers in the range [0,10),
 *  as long as they form a non-decreasing sequence.
 */
public class InOrder {
    public static void main (String[] args) {
        // Generate the two integers
        int num1 = (int) (Math.random()*(10));
        int num2 = (int) (Math.random()*(10));

        // Print the first random integer
        System.out.print(num1 + " ");

        // Generate and print subsequent random integers in a
non-decreasing sequence
        while (num2>=num1) {
            System.out.print(num2 + " ");
            num1 = num2;
            num2 = (int) (Math.random()*(9)) + 1;
        }
    }
}
```

DamkaBaord

```java
/**
 *  Gets a command-line argument n (int), and prints an n-by-n
damka board.
 */
public class DamkaBoard {
    public static void main(String[] args) {
    int num = Integer.parseInt(args[0]);
    for (int i = 0; i < num; i++) {
        if (i % 2 ==0) {
            for (int j = 0; j < num; j++) {
                System.out.print("* ");
            }
        }
        else {
            for (int j = 0; j < num; j++) {
                System.out.print(" *");
            }
    }
    System.out.println();
    }
}
}
```

Perfect

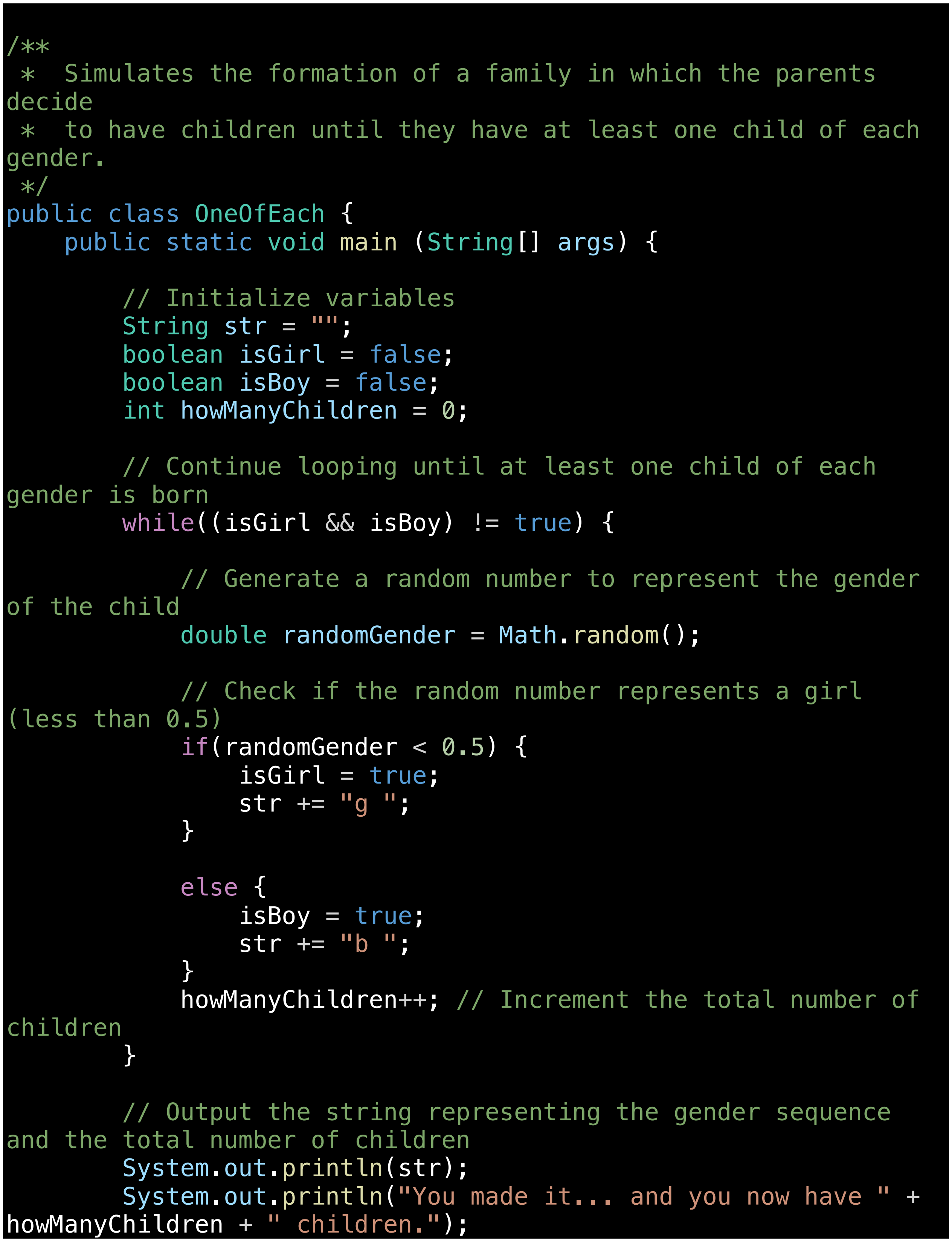
```java
/**
 *  Gets a command-line argument (int), and chekcs if the given
number is perfect.
 */
public class Perfect {
    public static void main(String[] args) throws Exception {

        // Parse the command-line argument as an integer and
declare some args
        int num = Integer.parseInt(args[0]);
        int sum = 1;
        String strPerfect = num + " is a perfect number since "
+ num + " = 1";

        // Iterate through potential divisors up to (num - 1)
and check if i is a divisor of num
        for (int i=2; i<num; i++) {
            if (num % i == 0) {
                sum += i;
                strPerfect = strPerfect + " + " + i;
            }
        }

        // Check if the sum of divisors equals the original
number
        if (sum == num) {
            System.out.print(strPerfect);
        }
        else {
            System.out.println(num + " is not a perfect
number");
        }
    }
}
```

OneOfEachStats



```java
/**
 *  Simulates the formation of a family in which the parents
decide
 *  to have children until they have at least one child of each
gender.
 */
public class OneOfEach {
    public static void main (String[] args) {

        // Initialize variables
        String str = "";
        boolean isGirl = false;
        boolean isBoy = false;
        int howManyChildren = 0;

        // Continue looping until at least one child of each
gender is born
        while((isGirl && isBoy) != true) {

            // Generate a random number to represent the gender
of the child
            double randomGender = Math.random();

            // Check if the random number represents a girl
(less than 0.5)
            if(randomGender < 0.5) {
                isGirl = true;
                str += "g ";
            }

            else {
                isBoy = true;
                str += "b ";
            }
            howManyChildren++; // Increment the total number of
children
        }

        // Output the string representing the gender sequence
and the total number of children
        System.out.println(str);
        System.out.println("You made it... and you now have " +
howManyChildren + " children.");
```

```
      }
}
```