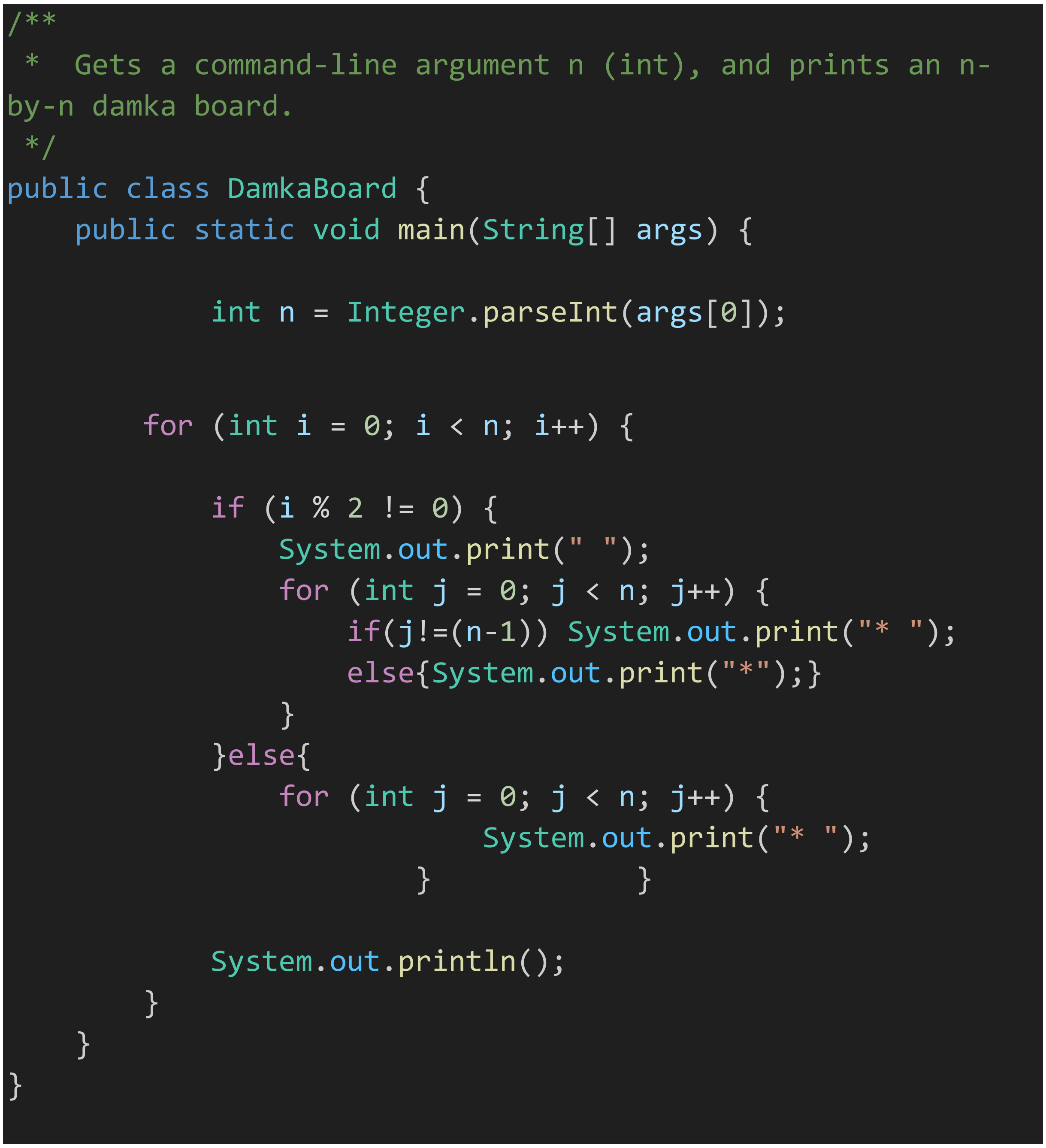
```java
/**
 *  Gets a command-line argument (int), and prints all the
divisors of the given number.
 */
 public class Divisors {
     public static void main (String[] args) {
        int a  = Integer.parseInt(args[0]);

        for (int i = 1; i <= a; i++)
        {
            if (a % i == 0)

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

```java
/**
 * 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){

            String input = args[0];

            String s = "";
            for (int i = input.length() - 1; i >= 0; i--){
                s = s + input.charAt(i);
            }
            System.out.println(s);

            int middleIndex = ((input.length() + 1) / 2) -1;
            char middle = input.charAt(middleIndex);
            System.out.println("The middle character is " +
middle);

    } }
```

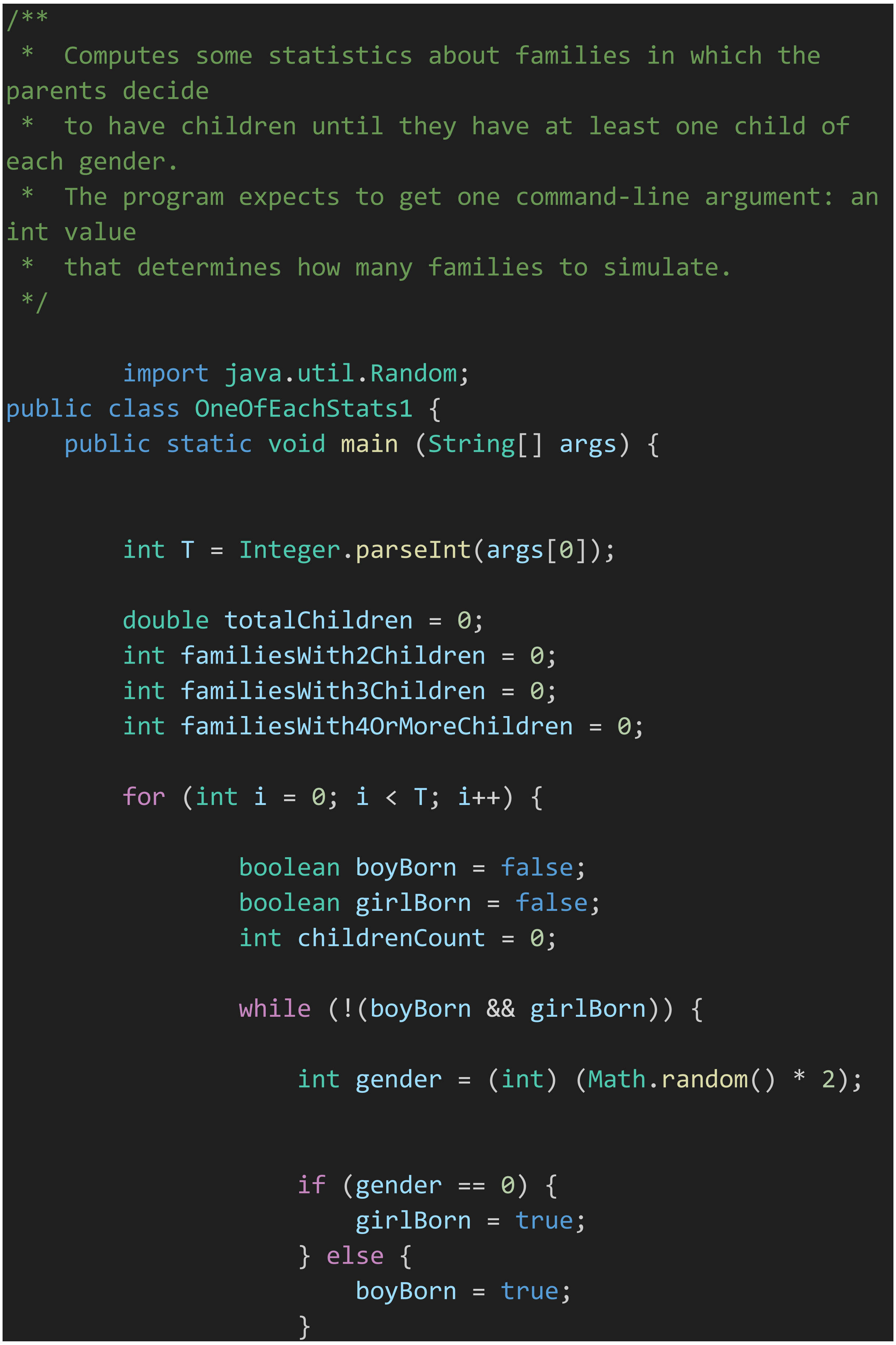
```java
import java.util.Random;

/**
 *  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) {

        Random random = new Random();
        int prevValue = -1;

        while (true) {
            int currentValue = random.nextInt(10);

            if (currentValue >= prevValue) {
                System.out.print(currentValue + " ");
                prevValue = currentValue;
            } else {
                break;
            }
        }

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

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

            int number = Integer.parseInt(args[0]);

            String a = number +" is a perfect number since "
+ number + " = 1";

            int sum = 1;
             for ( int i = 2; i < number - 1; i++)

             if (number%i == 0) {
                a = a + " + " + i;
                sum = sum + i;
            }
if (sum == number) {
    System.out.println(a);
        }
    else {
        System.out.println(number + " is not a perfect
number");

    } } }
```



```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 n = Integer.parseInt(args[0]);


        for (int i = 0; i < n; i++) {

            if (i % 2 != 0) {
                System.out.print(" ");
                for (int j = 0; j < n; j++) {
                    if(j!=(n-1)) System.out.print("* ");
                    else{System.out.print("*");}
                }
            }else{
                for (int j = 0; j < n; j++) {
                            System.out.print("* ");
                        }             }

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

```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) {

                boolean boyBorn = false;
                boolean girlBorn = false;
                int childrenCount = 0;

                while (!(boyBorn && girlBorn)) {

                    int gender = (int) (Math.random() * 2);

                    if (gender == 0) {
                        girlBorn = true;
                    } else {
                        boyBorn = true;
                    }

                    System.out.print((gender == 0) ? "g " :
"b ");


                    childrenCount++;
                }
                System.out.println();
                System.out.println("You made it... and you
now have " + childrenCount + " children.");
            }
        }
```



```java
/**
 *  Computes some statistics about families in which the
parents decide
 *  to have children until they have at least one child of
each gender.
 *  The program expects to get one command-line argument: an
int value
 *  that determines how many families to simulate.
 */

        import java.util.Random;
public class OneOfEachStats1 {
    public static void main (String[] args) {


        int T = Integer.parseInt(args[0]);

        double totalChildren = 0;
        int familiesWith2Children = 0;
        int familiesWith3Children = 0;
        int familiesWith4OrMoreChildren = 0;

        for (int i = 0; i < T; i++) {

                boolean boyBorn = false;
                boolean girlBorn = false;
                int childrenCount = 0;

                while (!(boyBorn && girlBorn)) {

                    int gender = (int) (Math.random() * 2);


                    if (gender == 0) {
                        girlBorn = true;
                    } else {
                        boyBorn = true;
                    }
                    childrenCount++;

                    totalChildren++;
                }

                if(childrenCount==2){
                    familiesWith2Children++;
                }
                if(childrenCount==3){
                    familiesWith3Children++;
                }
                if(childrenCount>3){
                    familiesWith4OrMoreChildren++;
                }

            }
                double x = totalChildren /T;
                System.out.println("Average: " + x + " children to get at least one of each gender.");
                System.out.println("Number of families with 2 children: " + familiesWith2Children);
                System.out.println("Number of families with 3 children: " + familiesWith3Children);
                System.out.println("Number of families with 4 or more children: " + familiesWith4OrMoreChildren);


        }

}
```

```java
import java.util.Random;
/**
 *  Computes some statistics about families in which the
parents decide
 *  to have children until they have at least one child of
each gender.
 *  The program expects to get two command-line arguments:
an int value
 *  that determines how many families to simulate, and an
int value
 *  that serves as the seed of the random numbers generated
by the program.
 *  Example usage: % java OneOfEachStats 1000 1
 */
public class OneOfEachStats {
    public static void main (String[] args) {

        int T = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);

        Random generator = new Random(seed);

        double totalChildren = 0;
        int familiesWith2Children = 0;
        int familiesWith3Children = 0;
        int familiesWith4OrMoreChildren = 0;

        for (int i = 0; i < T; i++) {

                boolean boyBorn = false;
                boolean girlBorn = false;
                int childrenCount = 0;

                while (!(boyBorn && girlBorn)) {

                    double gender = generator.nextDouble();;

                    if (gender <0.5) {
                        girlBorn = true;
                    } else {
                        boyBorn = true;
                    }

                    childrenCount++;

                    totalChildren++;
                }

                if(childrenCount==2){
                    familiesWith2Children++;
                }
                if(childrenCount==3){
                    familiesWith3Children++;
                }
                if(childrenCount>3){
                    familiesWith4OrMoreChildren++;
                }

            }
                double x = totalChildren /T;
                System.out.println("Average: " + x + "
children to get at least one of each gender.");
                System.out.println("Number of families with
2 children: " + familiesWith2Children);
                System.out.println("Number of families with
3 children: " + familiesWith3Children);
                System.out.println("Number of families with
4 or more children: " + familiesWith4OrMoreChildren);
                if(familiesWith2Children
>familiesWith3Children && familiesWith2Children
>familiesWith4OrMoreChildren )
                {
                    System.out.println("The most common
number of children is 2.");
                }
            if(familiesWith3Children >familiesWith2Children 
&& familiesWith3Children >familiesWith4OrMoreChildren )
                {
                    System.out.println("The most common 
number of children is 3.");
                }
                if(familiesWith4OrMoreChildren 
>familiesWith2Children &&  familiesWith4OrMoreChildren 
>familiesWith2Children )
                {
                    System.out.println("The most common 
number of children is 4 or more.");
                }
        }



    }
```