PATTERNS

1. 0-1 Triangle

```
public class bintriangle {
    public static void binarytriangle (int rows) {
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                if ((i + j)  % 2 == 0) {
                    System.out.print("1 ");
                } else {
                    System.out.print("0 ");
                }
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        binarytriangle(5);
    }
}
```

2. Character Pattern

```
public class characterpattern {
    public static void main(String[] args) {
        int n = 4;
        char ch = 'A';
        for (int line = 1; line <= n; line++) {
            for (int chars = 1; chars <= line; chars++) {
                System.out.print(ch);
                ch++;
            }
            System.out.println();
        }
    }
}
```

3. Floyd’s Triangle

```
import java.util.Scanner;
public class FloydsTriangle {
    public static void floydTriangle (int rows) {
        int counter = 1;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(counter + " ");
                counter++;
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int Rows = sc.nextInt();
        System.out.println("Floyd's Traingle is:");
        floydTriangle(Rows);
    }
}
```

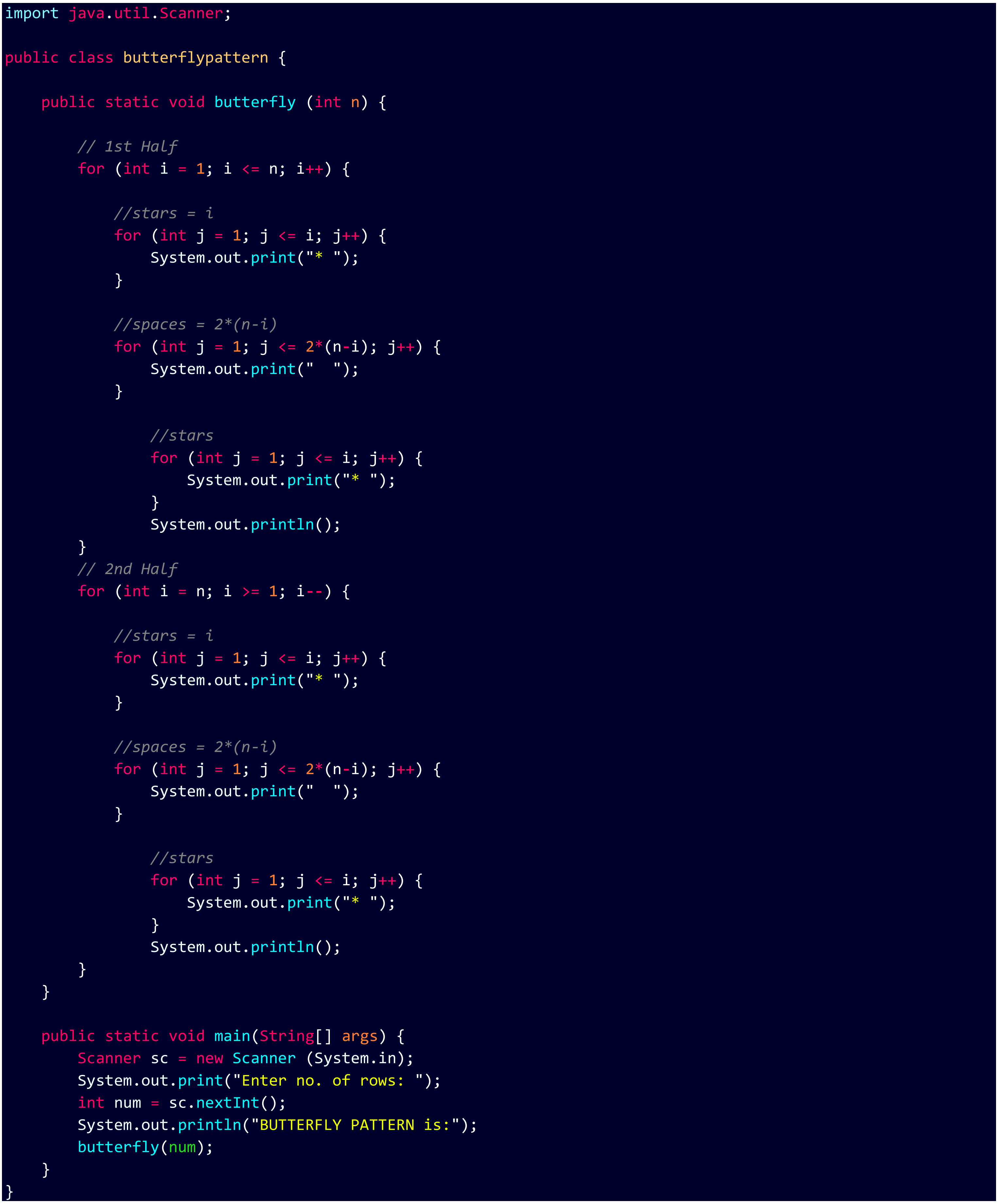
4. Diamond Pattern

```java
import java.util.Scanner;

public class diamondpattern {

    public static void diamond_pattern (int n) {
        for (int i = 1; i <= n ; i++) {
            for (int j = 1; j <= n-i; j++) {
                System.out.print("  ");
            }
            for (int j = 1; j <= (2*i)-1; j++) {
                System.out.print("* ");
            }
            for (int j = 1; j <= n-i; j++) {
                System.out.print("  ");
            }
            System.out.println();
        }
        for (int i = n; i >= 1; i--) {
            for (int j = 1; j <= n-i; j++) {
                System.out.print("  ");
            }
            for (int j = 1; j <= (2*i)-1; j++) {
                System.out.print("* ");
            }
            for (int j = 1; j <= n-i; j++) {
                System.out.print("  ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        System.out.println("DIAMOND PATTERN:");
        diamond_pattern(n);
    }
}
```

5. Half Pyramid Pattern

```java
public class halfpyramid {
    public static void main(String[] args) {
        for (int line = 1; line <= 4; line++) {
            for (int num = 1; num <=line; num++) {
                System.out.print(num + " ");
            }
            System.out.println();
        }
    }
}
```

6. Butterfly Pattern



```java
import java.util.Scanner;

public class butterflypattern {

    public static void butterfly (int n) {

        // 1st Half
        for (int i = 1; i <= n; i++) {

            //stars = i
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            //spaces = 2*(n-i)
            for (int j = 1; j <= 2*(n-i); j++) {
                System.out.print("  ");
            }

                //stars
                for (int j = 1; j <= i; j++) {
                    System.out.print("* ");
                }
                System.out.println();
        }
        // 2nd Half
        for (int i = n; i >= 1; i--) {

            //stars = i
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            //spaces = 2*(n-i)
            for (int j = 1; j <= 2*(n-i); j++) {
                System.out.print("  ");
            }

                //stars
                for (int j = 1; j <= i; j++) {
                    System.out.print("* ");
                }
                System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of rows: ");
        int num = sc.nextInt();
        System.out.println("BUTTERFLY PATTERN is:");
        butterfly(num);
    }
}
```

7. Hollow Rectangle

```java
import java.util.Scanner;

public class hollowrectangle {

    public static void hollow_rectangle(int totRows, int totCols) {
        for (int i = 1; i <= totRows; i++) {
            for (int j = 1; j <= totCols; j++) {
                if (i == 1 || i == totRows || j == 1 || j == totCols) {
                    System.out.print("* ");
                }
                else {
                    System.out.print("  ");
                }
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter no. of Cols: ");
        int cols = sc.nextInt();
        System.out.println("PATTERN is:");
        hollow_rectangle(rows, cols);
    }
}
```

8. Hollow Rhombus

```java
import java.util.Scanner;

public class hollowrhombus {
    public static void hollow_rhombus(int totrows) {

        // Spaces
        for (int i = 1; i <= totrows; i++) {
            for (int j = 1; j <= totrows - i; j++) {
                System.out.print("  ");
            }
            // Hollow Rectangle
            for (int j = 1; j <= totrows; j++) {
                if (i == 1 || i == totrows || j == 1 || j == totrows) {
                    System.out.print("* ");
                } else {
                    System.out.print("  ");
                }
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.println("Hollow Rhombus:");
        hollow_rhombus(rows);
    }
}
```

9. Inverted Half Pyramid

```
import java.util.Scanner;

public class invertedhalfpyramids {

    public static void half_num_pyramids (int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n-i+1; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.println("Inverted Half Pyramid is: ");
        half_num_pyramids(rows);
    }
}
```

10. Inverted Star Pattern

```
public class invertedstarpattern {
    public static void main(String[] args) {

        int n = 4;

        for (int line = 1; line <= n; line++) {
            for (int star = 1; star <= n-line+1; star++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

11. Number Pyramid Pattern

```
public class numberpyramidpattern {
    public static void num_pyramid_pattern (int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n-i; j++) {
                System.out.print(" ");
            }
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            for (int j = 1; j <= n-i; j++) {
                System.out.print(" ");
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        num_pyramid_pattern(6);
    }
}
```

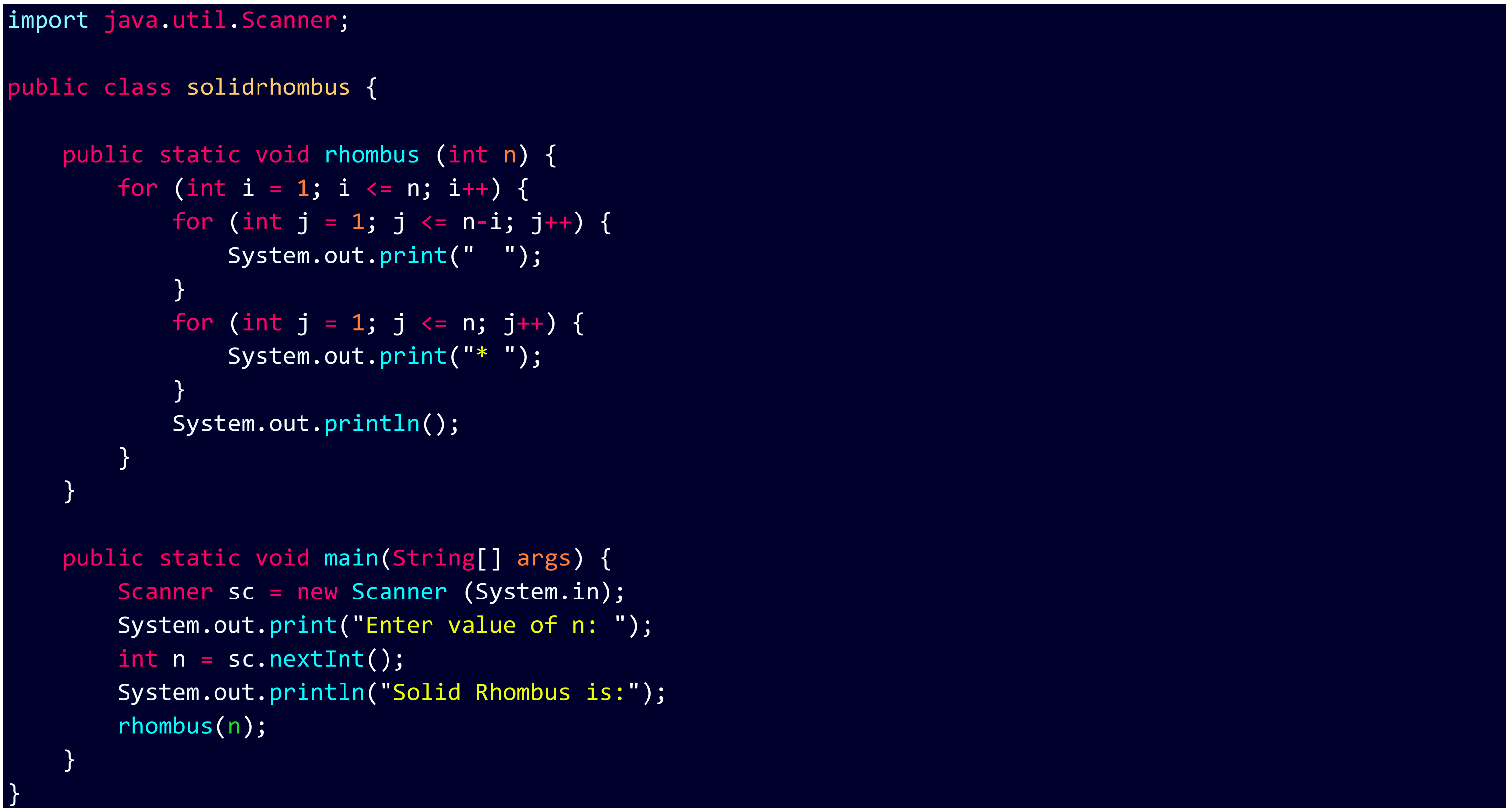
12. Palindrome Pattern

```java
import java.util.Scanner;

public class palindromepattern {

    public static void palindrome_pattern (int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n-i; j++) {
                System.out.print(" ");
            }
            for (int j = i; j >= 1; j--) {
                System.out.print(j);
            }
            for (int j = 2; j <= i; j++) {
                System.out.print(j);
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        System.out.println("PALINDROME PYRAMID:");
        palindrome_pattern(n);
    }
}
```

13. Inverted Half Pyramid

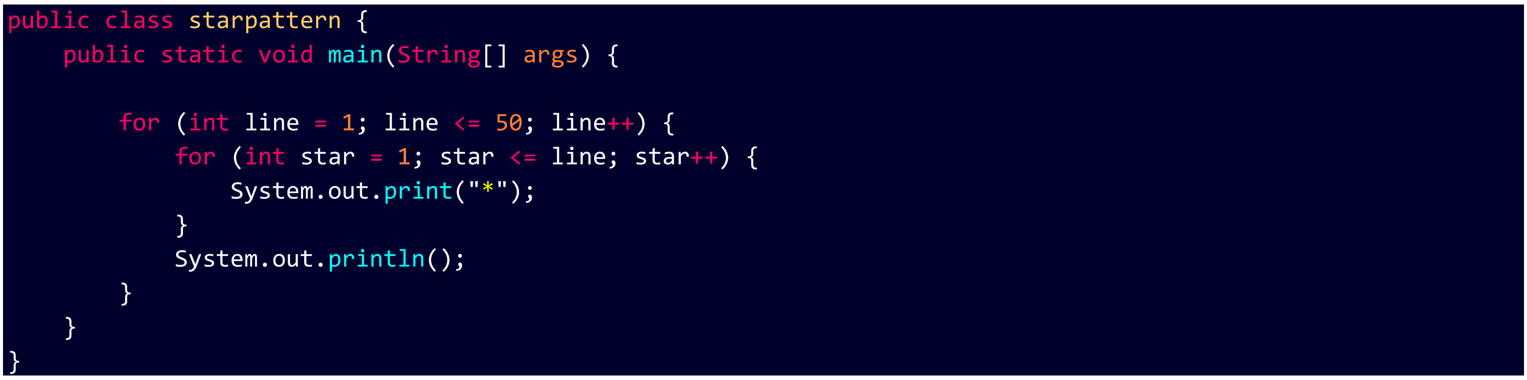
```java
import java.util.Scanner;

public class rotatedpyramid {

    public static void half_pyramid (int rows) {

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                System.out.print("  ");
            }
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Rows: ");
        int rows = sc.nextInt();
        System.out.println("INVERTED HALF PYRAMID:");
        half_pyramid(rows);
    }
}
```

14. Solid Rhombus



```
import java.util.Scanner;

public class solidrhombus {

    public static void rhombus (int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n-i; j++) {
                System.out.print("  ");
            }
            for (int j = 1; j <= n; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter value of n: ");
        int n = sc.nextInt();
        System.out.println("Solid Rhombus is:");
        rhombus(n);
    }
}
```

15. Star Pattern



```
public class starpattern {
    public static void main(String[] args) {

        for (int line = 1; line <= 50; line++) {
            for (int star = 1; star <= line; star++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```