Program Elegantly

PrintFace Example

Program Elegantly:

➢Choose Appropriate Names, Formats & Styles

➢Reveal Intentions and Computing Structures Visually

➢Make Simple Things Simple

➢Write Readable/Understandable Code

Look at a series of PrintFace Example Code:

```java
public class FacePrint0 {
        public static void main (String[] agrs) {
                    System.out.println( "     ///" );
                    System.out.println( " ///////" );
                    System.out.println( " |  o o  |" );
                    System.out.println( "@|   n   |@" );
                    System.out.println( " |  ~~~  |" );
                    System.out.println( "   -----" );
        }
}
```

```java
public class FacePrint1 {
    public static void main (String[] agrs) {
        System.out.println( "     ///" );
        System.out.println( " ///////" );
        System.out.println( " |  o o  |" );
        System.out.println( "@|   n   |@" );
        System.out.println( " |  ~~~  |" );
        System.out.println( "   -----" );
    }
}
```
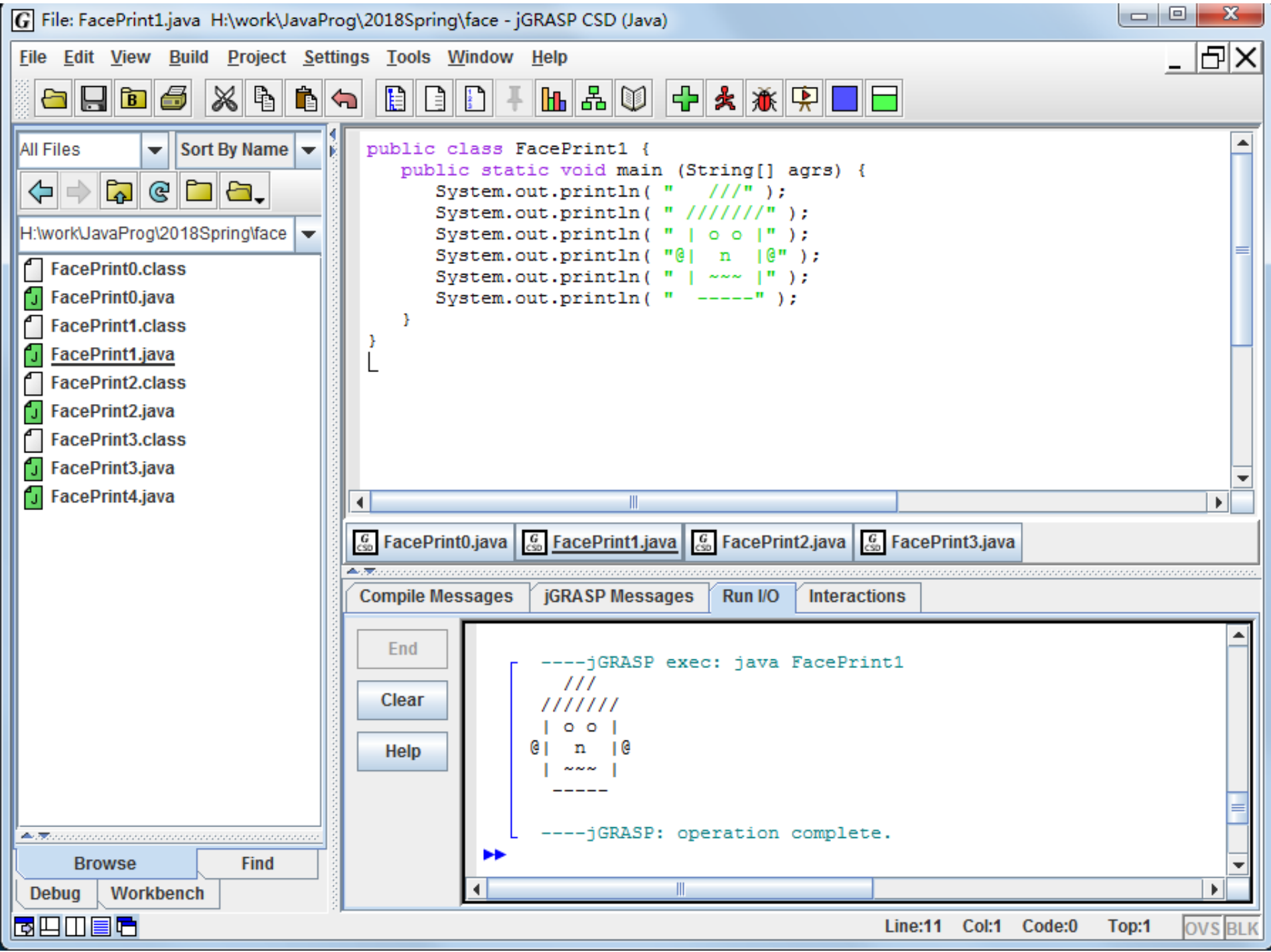
Researches Showed that 2~4 blanks are best readable indentations!

All Files

Sort By Name

H:\work\JavaProg\2018Spring\face

- FacePrint0.class
- FacePrint0.java
- FacePrint1.class
- FacePrint1.java
- FacePrint2.class
- FacePrint2.java
- FacePrint3.class
- FacePrint3.java
- FacePrint4.java

```java
public class FacePrint1 {
    public static void main (String[] agrs) {
        System.out.println( "   ///" );
        System.out.println( " ///////" );
        System.out.println( " | o o |" );
        System.out.println( "@|   n  |@" );
        System.out.println( " | ~~~ |" );
        System.out.println( "  -----" );
    }
}
```

FacePrint0.java   FacePrint1.java   FacePrint2.java   FacePrint3.java

Compile Messages | jGRASP Messages | Run I/O | Interactions

End

Clear

Help

```
    ----jGRASP exec: java FacePrint1
       ///
     ///////
     | o o |
    @|   n  |@
     | ~~~ |
      -----

    ----jGRASP: operation complete.
```

Browse   Find

Debug   Workbench

Line:11 Col:1 Code:0 Top:1   OVS BLK

```java
public class FacePrint1 {
    public static void main (String[] agrs) {
        System.out.println( "    ///" );
        System.out.println( " ///////" );
        System.out.println( " | o o |" );
        System.out.println( "@|   n   |@" );
        System.out.println( " |  ~~~  |" );
        System.out.println( "   -----" );
    }
}
```

```java
public class FacePrint2 {
    public static void main (String[] agrs) {
        System.out.print(
            "    ///\n" +
            " ///////\n" +
            " | o o |\n" +
            "@|   n   |@\n" +
            " |  ~~~  |\n" +
            "   -----\n"
        );
    }
}
```

Remove Duplications whenever possible.

```java
public class FacePrint2 {
    public static void main (String[] agrs) {
        System.out.print(
            "     ///\n" +
            " ///////\n" +
            " |  o  o  |\n" +
            "@|     n    |@\n" +
            " |   ~~~   |\n" +
            "     -----\n"
        );
    }
}
```

```java
public class FacePrint3 {
    public static void main (String[] agrs) {
        String face =
            "     ///\n" +
            " ///////\n" +
            " |  o  o  |\n" +
            "@|     n    |@\n" +
            " |   ~~~   |\n" +
            "     -----\n" ;
        System.out.print( face );
    }
}
```

Naming something by a variable helps understanding.

```java
public class FacePrint3 {
    public static void main (String[] agrs) {
        String face =
            "   ///\n" +
            " ///////\n" +
            " | o o |\n" +
            "@|   n   |@\n" +
            " | ~~~ |\n" +
            "   -----\n" ;
        System.out.print( face );
    }
}
```

```java
public class FacePrint4 {
    public static void main (String[] agrs) {
        String face = "   ///\n ///////\n | o o |\n@|   n   |@\n | ~~~ |\n   -----\n" ;
        System.out.print( face );
    }
}
```

Programming for Humans First, Then for Compilers.

```
1 x = a+b*c+d;
2
3 x = a + b * c + d;
4
5 x = a + b*c + d;
6
7 x = a+b * c+d;
8
```

➢ Crowded, Bad

➢ Clear, Good

➢ Creative, Best

➢ Confused, Worse

✓ Let the code speaks itself.

✓ The Layout Structure of code reveals the Motivations / Logic Structure / Meaning Group (代码布局结构揭示计算的意图/逻辑结构/意群结构)

✓ Help to understand and debug , 节省思维

Program Elegantly:

➤Choose Appropriate Names, Formats & Styles

➤Reveal Intentions and Computing Structures Visually

➤Make Simple Things Simple

➤Write Readable/Understandable Code