Demo.java

```java
/**
 * Demo is a book testing class
 *
 * <pre>
 * book1
 * number of pages: 200
 * width: 3
 * height: 4
 * number of lines on the front cover: 5
 * number of lines on the back cover: 6
 *
 * book2
 * number of pages: 300
 * width: 4
 * height: 4
 * number of lines on the front cover: 4
 * number of lines on the back cover: 5
 *
 * book3
 * number of pages: 435
 * width: 5
 * height: 7
 * number of lines on the front cover: 5
 * number of lines on the back cover: 6
 * </pre>
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1.1 Nov 11 2020
 */
public class Demo {
    public static void main(String argv[]) {

        try {
            Cover f1 = new Cover(5);
            Cover b1 = new Cover(6);
            Book book1 = new Book(200, 3, 4, f1, b1);
        } catch (EmptyBook n) {
            System.out.println("Error code " + n.getCode() + ": "
                    + "Number of pages of the book is lees or equal tha
n 0 " + n.getNp());
        } catch (SquareBook n) {
            System.out.println("Error code " + n.getCode() + ": " + "Th
e book's width is equal to its height. Side is: "
```

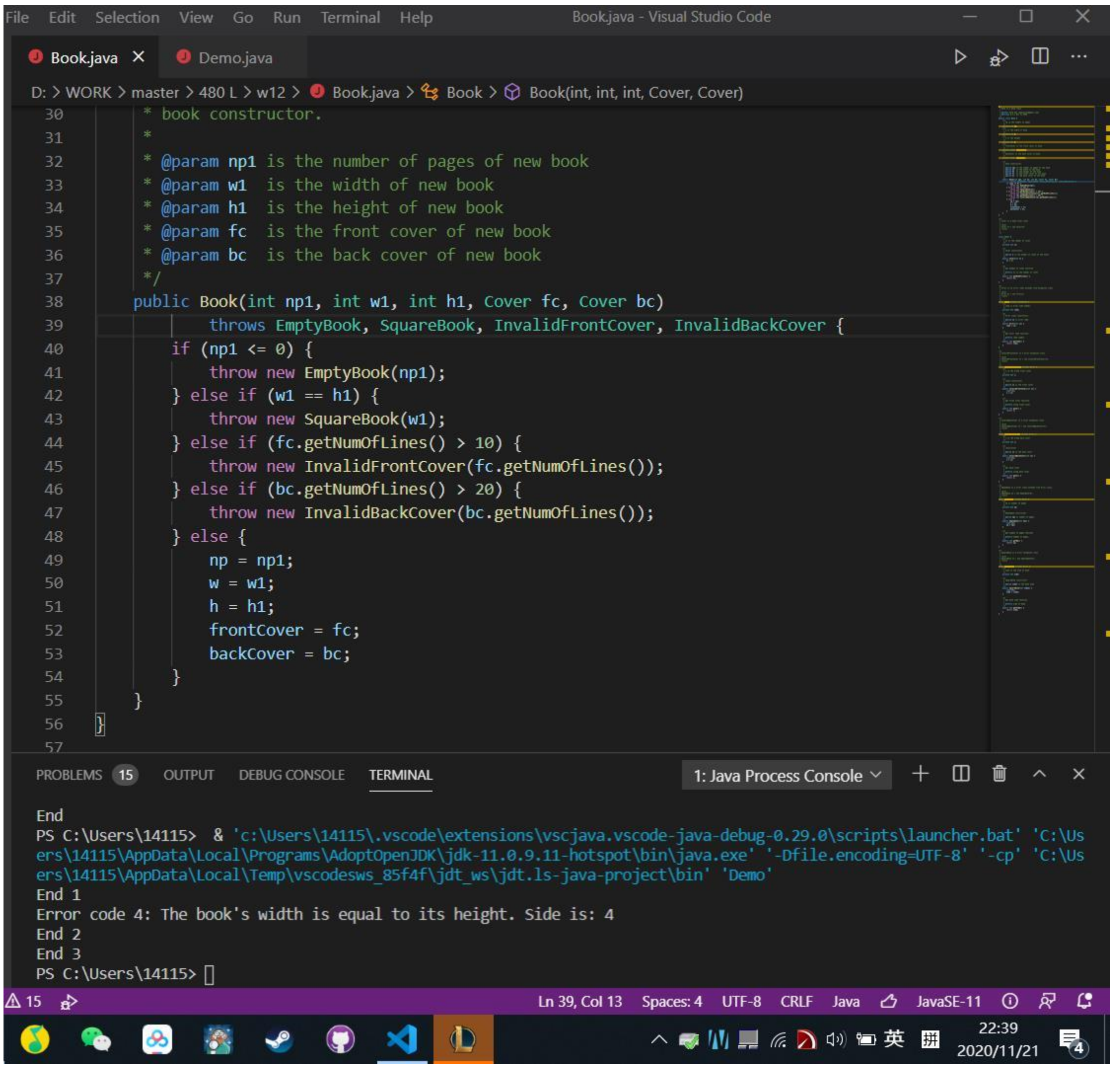
```
                    + n.getSide());
        } catch (InvalidFrontCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for front cover is more than 10 " + n.getI());
        } catch (InvalidBackCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for back cover is more than 20  " + n.getI());
        } finally {
            System.out.println("End 1");
        }

        try {
            Cover f2 = new Cover(4);
            Cover b2 = new Cover(5);
            Book book2 = new Book(300, 4, 4, f2, b2);
        } catch (EmptyBook n) {
            System.out.println("Error code " + n.getCode() + ": "
                    + "Number of pages of the book is lees or equal tha
n 0 " + n.getNp());
        } catch (SquareBook n) {
            System.out.println("Error code " + n.getCode() + ": " + "Th
e book's width is equal to its height. Side is: "
                    + n.getSide());
        } catch (InvalidFrontCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for front cover is more than 10 " + n.getI());
        } catch (InvalidBackCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for back cover is more than 20  " + n.getI());
        } finally {
            System.out.println("End 2");
        }

        try {
            Cover f3 = new Cover(5);
            Cover b3 = new Cover(6);
            Book book3 = new Book(435, 5, 7, f3, b3);
        } catch (EmptyBook n) {
            System.out.println("Error code " + n.getCode() + ": "
                    + "Number of pages of the book is lees or equal tha
n 0 " + n.getNp());
        } catch (SquareBook n) {
            System.out.println("Error code " + n.getCode() + ": " + "Th
e book's width is equal to its height. Side is: "
                    + n.getSide());
        } catch (InvalidFrontCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for front cover is more than 10 " + n.getI());
        } catch (InvalidBackCover n) {
            System.out.println(
                    "Error code " + n.getCode() + ": " + "Number of lin
es for back cover is more than 20  " + n.getI());
        } finally {
            System.out.println("End 3");
        }
    }
}
```

Book.java

```
/**
 * Book is a book class
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1.1 Nov 11 2020
 */
public class Book {
    /**
     * np is the number of pages
     */
    private int np;
    /**
     * w is the width of book
     */
    private int w;
    /**
     * h is the height
     */
    private int h;
    /**
     * frontCover is the front cover of book
     */
    private Cover frontCover;
    /**
     * backCover is the back cover of book
     */
    private Cover backCover;

    /**
     * book constructor.
     *
     * @param np1 is the number of pages of new book
     * @param w1  is the width of new book
     * @param h1  is the height of new book
     * @param fc  is the front cover of new book
     * @param bc  is the back cover of new book
     */
    public Book(int np1, int w1, int h1, Cover fc, Cover bc)
            throws EmptyBook, SquareBook, InvalidFrontCover, InvalidBac
kCover {
        if (np1 <= 0) {
            throw new EmptyBook(np1);
        } else if (w1 == h1) {
            throw new SquareBook(w1);
        } else if (fc.getNumOfLines() > 10) {
            throw new InvalidFrontCover(fc.getNumOfLines());
        } else if (bc.getNumOfLines() > 20) {
            throw new InvalidBackCover(bc.getNumOfLines());
        } else {
            np = np1;
            w = w1;
            h = h1;
            frontCover = fc;
            backCover = bc;
        }
    }
}

/**
 * Cover is a book cover class
 *
 * <pre>
 * Cover f1 = new Cover(5);
 * </pre>
 *
 */
```

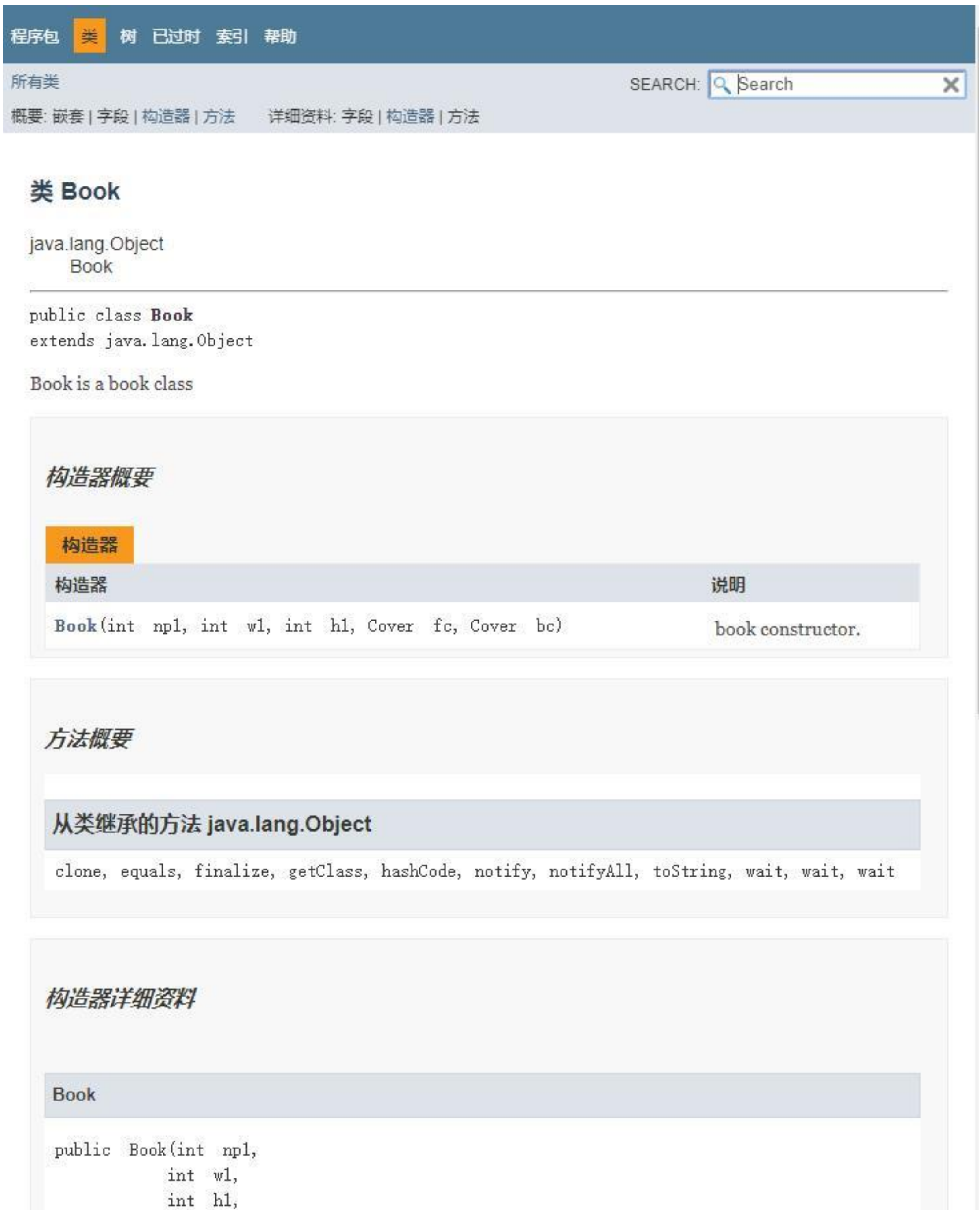
```java
class Cover {
    /**
     * nl is the number of lines
     */
    private int nl;

    /**
     * Cover constructor
     *
     * @param n is the number of lines of new Cover
     */
    public Cover(int n) {
        nl = n;
    }

    /**
     * get number of lines function
     *
     * @return nl is the number of lines
     */
    public int getNumOfLines() {
        return nl;
    }
}

/**
 * Error is an error class extends from Exception class
 *
 * <pre>
 * Error e1 = new Error(1)
 * </pre>
 *
 */
class Error extends Exception {
    /**
     * code is error code number.
     */
    private int code;

    /**
     * Error class constructor.
     *
     * @param i1 is error code.
     */
    public Error(int i1) {
        code = i1;
    }

    /**
     * get error code function.
     *
     * @return code number.
     */
    public int getCode() {
        return code;
    }
}

/**
 * InvalidFrontCover is a error exception class
 *
 * <pre>
 * InvalidFrontCover f1 = new InvalidFrontCover(5);
 * </pre>
 *
 */
class InvalidFrontCover extends Error {
    /**
     * i is the wrong front cover
     */
    private int i;

    /**
     * class constructor
     *
     * @param i1 is the front cover
     */
    public InvalidFrontCover(int i1) {
        super(1);
        i = i1;
    }

    /**
     * get front cover function
     *
     * @return wrong front cover
     */
    public int getI() {
        return i;
    }
}

/**
 * InvalidBackCover is a error exception class
 *
 * <pre>
 * InvalidBackCover f1 = new InvalidBackCover(5);
 * </pre>
 *
 */
class InvalidBackCover extends Error {
    /**
     * i is the wrong back cover
     */
    private int i;

    /**
     * constructor
     *
     * @param i1 is the back cover
     */
    public InvalidBackCover(int i1) {
        super(2);
        i = i1;
    }

    /**
     * get back cover
     *
     * @return wrong back cover
     */
    public int getI() {
        return i;
    }
}

/**
 * EmptyBook is a error class extends from Error class.
 *
 * <pre>
 * EmptyBook e1 = new EmptyBook(5);
 * </pre>
 */
class EmptyBook extends Error {
    /**
     * np is number of pages
     */
    private int np;

    /**
     * EmptyBook constructor.
     *
     * @param np1 is number of pages.
     */
    public EmptyBook(int np1) {
        super(3);
        np = np1;
    }

    /**
     * get number of pages function
     *
     * @return number of pages.
     */
    public int getNp() {
        return np;
    }
}

/**
 * SquareBook is a error exception class
 *
 * <pre>
 * SquareBook s1 = new SquareBook(5);
 * </pre>
 *
 */
class SquareBook extends Error {
    /**
     * side is the side of book
     */
    private int side;

    /**
     * SquareBook constructor
     *
     * @param side1 is the book side
     */
    public SquareBook(int side1) {
        super(4);
        side = side1;
    }

    /**
     * get book side function
     *
     * @return side of book
     */
    public int getSide() {
        return side;
    }
}
```

The screen shoot of output and Javadoc


File Edit Selection View Go Run Terminal Help
Book.java - Visual Studio Code
Book.java
Demo.java
D: > WORK > master > 480 L > w12 > Book.java > Book > Book(int, int, int, Cover, Cover)
PROBLEMS 15
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Java Process Console
Ln 39, Col 13 Spaces: 4 UTF-8 CRLF Java JavaSE-11
22:39
2020/11/21

```java
     * book constructor.
     *
     * @param np1 is the number of pages of new book
     * @param w1  is the width of new book
     * @param h1  is the height of new book
     * @param fc  is the front cover of new book
     * @param bc  is the back cover of new book
     */
    public Book(int np1, int w1, int h1, Cover fc, Cover bc)
            throws EmptyBook, SquareBook, InvalidFrontCover, InvalidBackCover {
        if (np1 <= 0) {
            throw new EmptyBook(np1);
        } else if (w1 == h1) {
            throw new SquareBook(w1);
        } else if (fc.getNumOfLines() > 10) {
            throw new InvalidFrontCover(fc.getNumOfLines());
        } else if (bc.getNumOfLines() > 20) {
            throw new InvalidBackCover(bc.getNumOfLines());
        } else {
            np = np1;
            w = w1;
            h = h1;
            frontCover = fc;
            backCover = bc;
        }
    }
}
```

```
End
PS C:\Users\14115>  & 'c:\Users\14115\.vscode\extensions\vscjava.vscode-java-debug-0.29.0\scripts\launcher.bat' 'C:\Us
ers\14115\AppData\Local\Programs\AdoptOpenJDK\jdk-11.0.9.11-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Us
ers\14115\AppData\Local\Temp\vscodesws_85f4f\jdt_ws\jdt.ls-java-project\bin' 'Demo'
End 1
Error code 4: The book's width is equal to its height. Side is: 4
End 2
End 3
PS C:\Users\14115> 
```


程序包 类 树 已过时 索引 帮助

所有类

SEARCH: Search

概要: 嵌套 | 字段 | 构造器 | 方法   详细资料: 字段 | 构造器 | 方法

类 Book

java.lang.Object
    Book

```
public class Book
extends java.lang.Object
```

Book is a book class

构造器概要

构造器

构造器	说明
`Book(int  np1, int  w1, int  h1, Cover  fc, Cover  bc)`	book constructor.

方法概要

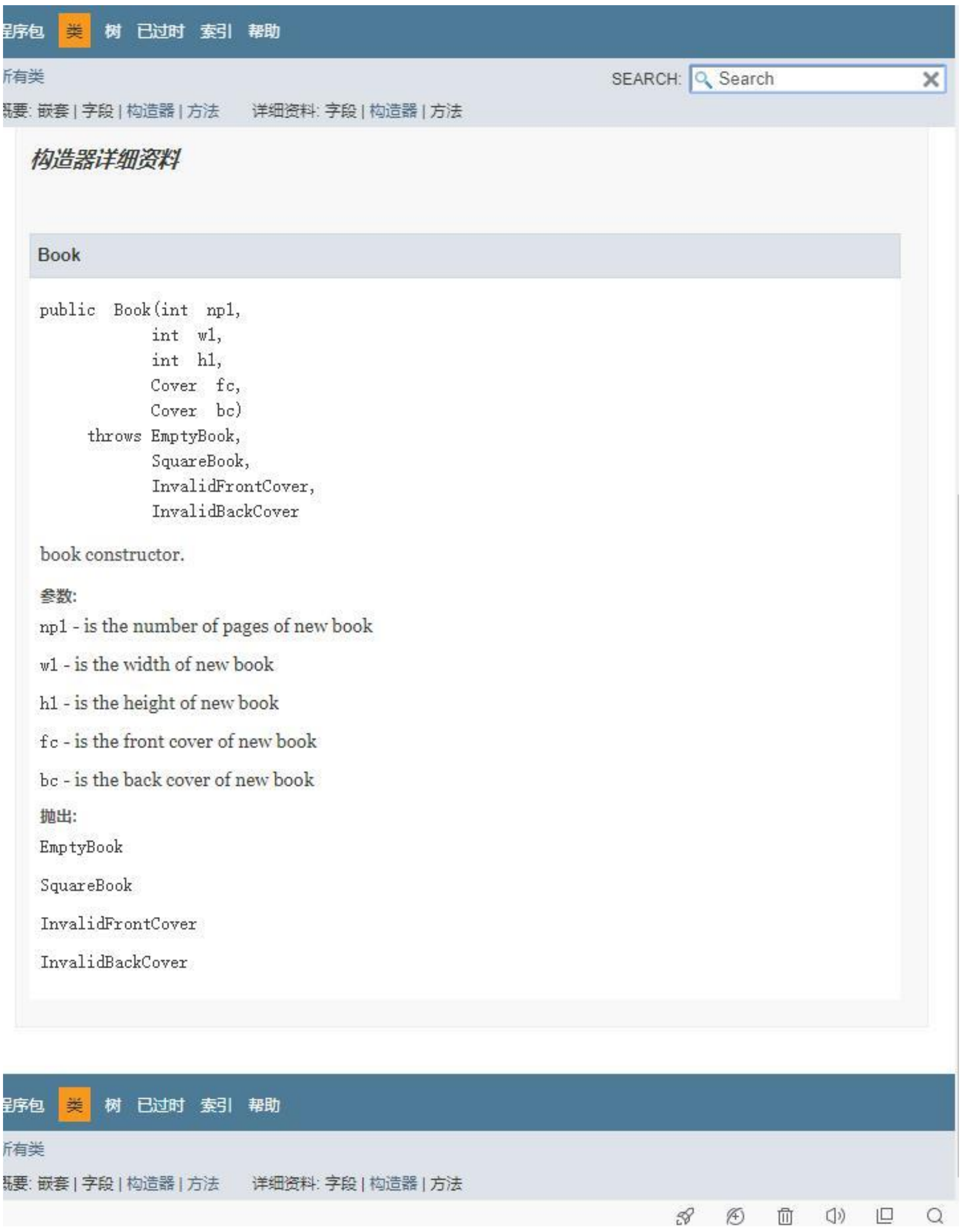
从类继承的方法 java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

构造器详细资料

Book

```
public  Book(int  np1,
             int  w1,
             int  h1,
```



星序包 类 树 已过时 索引 帮助
所有类
SEARCH: Search
概要: 嵌套 | 字段 | 构造器 | 方法 详细资料: 字段 | 构造器 | 方法

构造器详细资料

Book

```
public  Book(int  np1,
             int  w1,
             int  h1,
             Cover  fc,
             Cover  bc)
     throws EmptyBook,
             SquareBook,
             InvalidFrontCover,
             InvalidBackCover
```

book constructor.

参数:

`np1` - is the number of pages of new book

`w1` - is the width of new book

`h1` - is the height of new book

`fc` - is the front cover of new book

`bc` - is the back cover of new book

抛出:

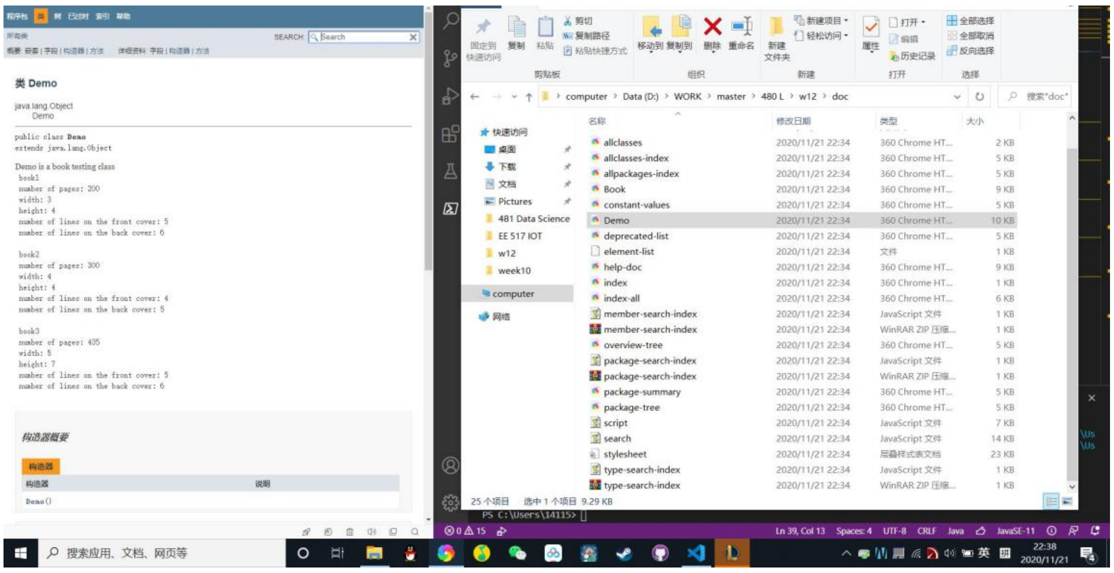
`EmptyBook`

`SquareBook`

`InvalidFrontCover`

`InvalidBackCover`

星序包 类 树 已过时 索引 帮助
所有类
概要: 嵌套 | 字段 | 构造器 | 方法 详细资料: 字段 | 构造器 | 方法


程序包 类 树 已过时 索引 帮助
所有类
SEARCH: Search
概要: 嵌套 | 字段 | 构造器 | 方法 详细资料: 字段 | 构造器 | 方法

类 Demo

java.lang.Object
  Demo

```
public class Demo
extends java.lang.Object
```

Demo is a book testing class

```
 book1
 number of pages: 200
 width: 3
 height: 4
 number of lines on the front cover: 5
 number of lines on the back cover: 6

 book2
 number of pages: 300
 width: 4
 height: 4
 number of lines on the front cover: 4
 number of lines on the back cover: 5

 book3
 number of pages: 435
 width: 5
 height: 7
 number of lines on the front cover: 5
 number of lines on the back cover: 6
```

构造器概要

构造器

构造器	说明
Demo()	

computer › Data (D:) › WORK › master › 480 L › w12 › doc

名称	修改日期	类型	大小
allclasses	2020/11/21 22:34	360 Chrome HT...	2 KB
allclasses-index	2020/11/21 22:34	360 Chrome HT...	5 KB
allpackages-index	2020/11/21 22:34	360 Chrome HT...	5 KB
Book	2020/11/21 22:34	360 Chrome HT...	9 KB
constant-values	2020/11/21 22:34	360 Chrome HT...	5 KB
Demo	2020/11/21 22:34	360 Chrome HT...	10 KB
deprecated-list	2020/11/21 22:34	360 Chrome HT...	5 KB
element-list	2020/11/21 22:34	文件	1 KB
help-doc	2020/11/21 22:34	360 Chrome HT...	9 KB
index	2020/11/21 22:34	360 Chrome HT...	1 KB
index-all	2020/11/21 22:34	360 Chrome HT...	6 KB
member-search-index	2020/11/21 22:34	JavaScript 文件	1 KB
member-search-index	2020/11/21 22:34	WinRAR ZIP 压缩...	1 KB
overview-tree	2020/11/21 22:34	360 Chrome HT...	5 KB
package-search-index	2020/11/21 22:34	JavaScript 文件	1 KB
package-search-index	2020/11/21 22:34	WinRAR ZIP 压缩...	1 KB
package-summary	2020/11/21 22:34	360 Chrome HT...	5 KB
package-tree	2020/11/21 22:34	360 Chrome HT...	5 KB
script	2020/11/21 22:34	JavaScript 文件	7 KB
search	2020/11/21 22:34	JavaScript 文件	14 KB
stylesheet	2020/11/21 22:34	层叠样式表文档	23 KB
type-search-index	2020/11/21 22:34	JavaScript 文件	1 KB
type-search-index	2020/11/21 22:34	WinRAR ZIP 压缩...	1 KB

25 个项目 选中 1 个项目 9.29 KB

```
PS C:\Users\14115>
```