一.异常:Java:有问题的代码不能运行

问题(红了)

```
1.错误(Error)
2.异常(Exception)
```

解决错的误方式:修改代码

解决异常的方式:

```
1.自行处理 try...catch
2.继续抛出异常:
```

```
System.out.println("123");
Thread.sleep(123);
```

为什么要出现错误?

```
Java的编译器或运行环境给你反馈
```

```
int a1[]=new int[10];
int a2[]={1,2,3};
int a3[]={1};
int []a4=null;
```

为什么要出现异常的?

```
System.out.println(st(a1));
System.out.println(st(a2));
System.out.println(st(a3));

```

```
try {
//try有可能出现异常的代码
System.out.println(st(a4));
}catch (NoSuchElementException e){
System.out.println(e.getMessage());
}catch (NullPointerException e){
//当出现异常会执行
//catch用于捕获异常对象
System.out.println(e.getMessage());
 }finally {
 //永远都会执行
 System.out.println("finally");
 }
```

```
eg: try {
```

```
st(a3);
} catch (Exception e) {
e.printStackTrace();
}
```

二.throw和throws的区别

throw用在方法体中;throws用在方法声明(定义)中

throw抛出异常对象;throws抛出异常类型

三.异常的分类:

1.运行时异常(非检查性异常):继承于RuntimeException的异常,不用强制性要求,对异常进行处理

2.非运行时异常(检查型异常)除运行时异常以外的异常.必须异常处理

final:用于修饰变量,属性,方法,类型.

finally:使用try...catch....finally结构的 一部分,finally中的代码肯定会执行

finalize:object类中的方法,析构方法,于销毁对象

四.日志管理

1.收集项目运行问题

2.调试代码

打印:影响项目的执行,抢了服务器资源

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

五.日志框架

log4j的组成:

1.日志记录器:输出日志

2.输出端:指定日志输出的位置

3.日志格式化:控制日志输出的样式

方式一：

创建日志记录器

```
Logger logger=Logger.getLogger(Main.class);
```

日志格式

1.简单格式()

```
SimpleLayout simpleLayout=new SimpleLayout();
```

2.HTML格式

```
HTMLLayout htmlLayout=new HTMLLayout();
```

3.自定义格式(txt)

```
https://www.yiibai.com/log4j/log4j_patternlayout.html
PatternLayout patternLayout=new PatternLayout("%r [%t] %-5p %c - %m%n");
```

创建输出端

1.控制台

```
ConsoleAppender consoleAppender=new ConsoleAppender(simpleLayout);
logger.addAppender(consoleAppender);
```

2.文件

```
FileAppender fileAppender=new FileAppender(htmlLayout,"log.html");
logger.addAppender(fileAppender);
```

3.定期生成文件

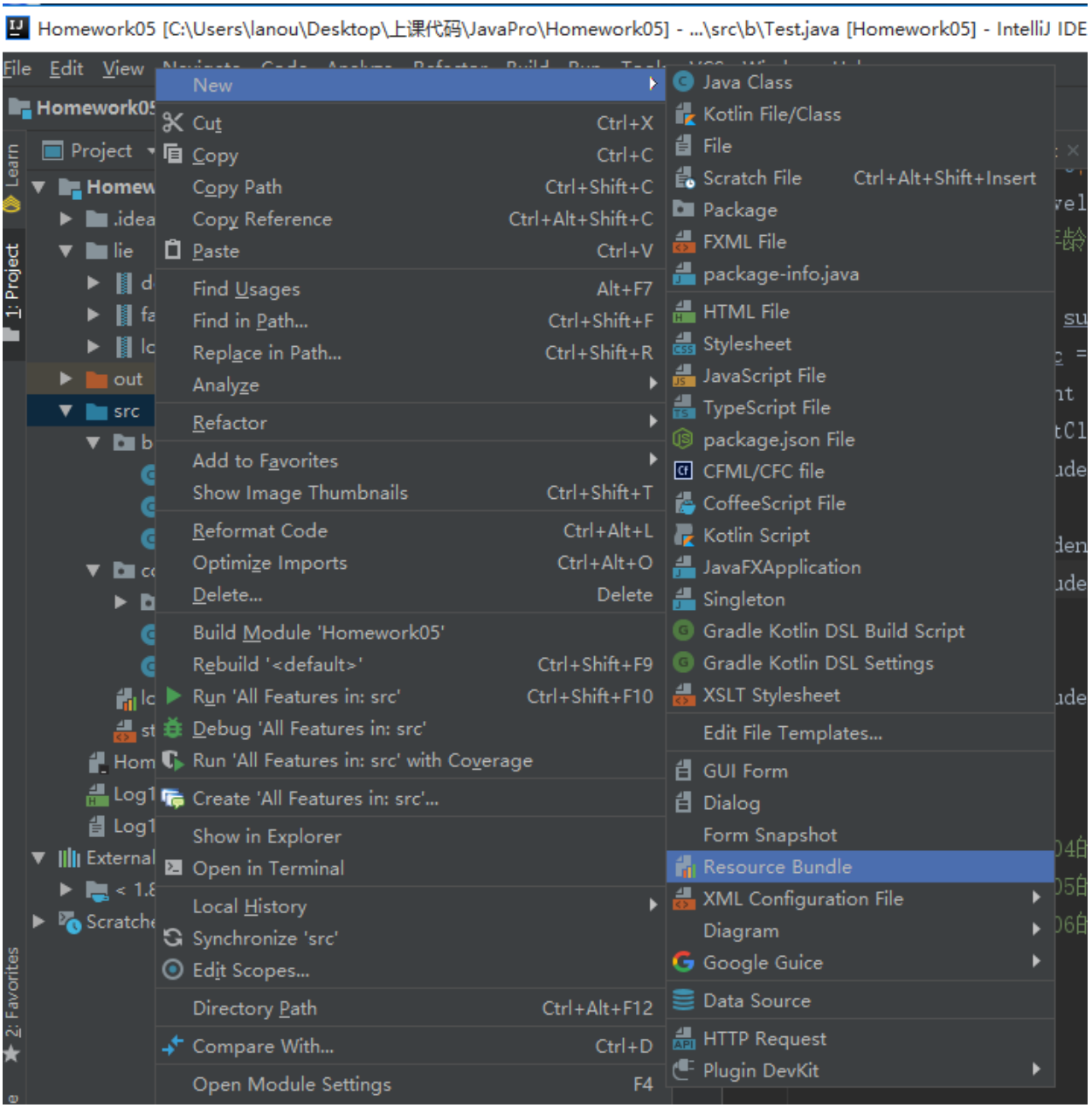
```
DailyRollingFileAppender dailyRollingFileAppender=
new DailyRollingFileAppender(patternLayout,"log.txt","yyyy-MM-dd");
logger.addAppender(dailyRollingFileAppender);
```

设置日志等级

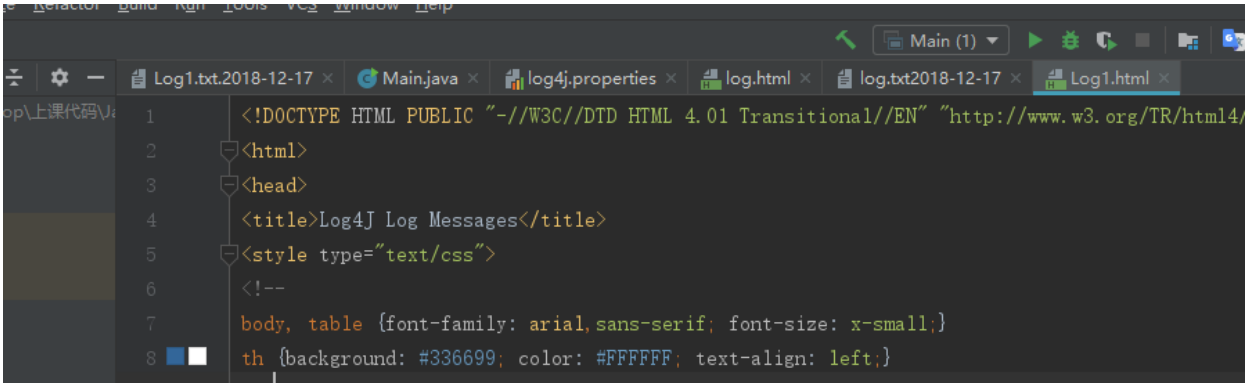
```
logger.setLevel(Level.INFO);
```

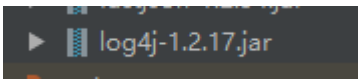
使用日志记录打印日志

```
logger.trace("123");
logger.debug("123");
logger.info("123");
logger.error("123");
logger.fatal("123");
```



方法:在scr下新建一个日志,复制下面的东西.html和txt文本.





```
#日志级别加输出端
log4j.rootLogger=INFO,a,b,c
#控制台
log4j.appender.a=org.apache.log4j.ConsoleAppender
log4j.appender.a.layout=org.apache.log4j.SimpleLayout

#文件
log4j.appender.b=org.apache.log4j.FileAppender
log4j.appender.b.layout=org.apache.log4j.HTMLLayout
 log4j.appender.b.File=Log1.html
 #定期生成文件
 log4j.appender.c=org.apache.log4j.DailyRollingFileAppender
 log4j.appender.c.layout=org.apache.log4j.PatternLayout
 log4j.appender.c.layout.ConversionPattern=%m%n
 log4j.appender.c.File=Log1.txt
```