



# Java Lab5: 报错了一快跑！

- 🚨 怎么办
- 📖 题目讲解
- </> 杂项



# ! 怎么办

人和代码，有一个能跑就行。

- 你有一份多到写不完的Java作业；
- 今天就是DDL，而且还不是晚上十二点；
- 助教还不好通融，总打哑谜；
- 今天是**星期四**，你的代码**疯狂**报错：

```
Exception in thread "main" java.lang.NoMoneyException: KFC Crazy Thursday V Me 50$.
```

- 而你，只是一个ECNU的CS大一学生。

# ! 怎么办

人和代码，有一个能跑就行。

- 首先，电脑不会爆炸，不需担心，不用跑。
- 其次，想象你是一个程序的设计者，如果你的程序在运行时出错，你是否要留下一些线索以帮助使用者找到问题所在？
- **报错信息并不是毫无意义的 除了MySQL**
- 尝试读一下Java给你的报错信息，会给你很多帮助。

例如在运行时：

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: row must be between 0 and 297
    at Picture.set(Picture.java:263)
    at Main.main(Main.java:18)
```

- 为什么报错？
- 错误发生在哪里？

# ! 怎么办

人和代码，有一个能跑就行。

- 现在你知道为什么报错了，也知道在哪里出错了，也许可以尝试修复bug。
- RTFM, STFW

如果没有报错，但是结果不对怎么办？

- 很明显你犯错的概率比电脑犯错的概率要大得多，所以请检查你的代码。
- **逻辑错误和未定义行为**是常见的类型。
- RTFM，我不要你以为，我要文档以为。

- 编译的时候报Warning怎么办？

```
$ javac Picture.java
```

注：Picutre.java使用或覆盖了已过时的API。

注：有关详细信息，请使用 `-Xlint:deprecation` 重新编译

首先，如果报Warning，可能是并不影响使用的，但并不保证。

看到那个提示了吗，按照提示重新编译，然后STFW, RTFM。

# 题目讲解

*you want it, you split() it.*

给定一个网址字符串, 根据 "." 将它分解成字符串. 例如 `http://www.ecnu.edu.cn` 分解成为 `http://www`, `ecnu`, `edu`, `cn`. (提示: 使用 `split()` 方法)

- EEEEEasy, 按照提示使用 `split()` 方法:

```
public class SplitString {  
    public static void main(String []args)  
    {  
        String []str=args[0].split(".");  
        for(String temp:str) System.out.println(temp);  
    }  
}
```

- 寄, 怎么没用

1. 电脑不会犯错

2. RTFM,STFW

# 题目讲解

*you want it, you speculate it.*

给定一个字符串, 代表一个16进制数. 将其转换成10进制整数, 输出到标准输出.

偷鸡:

```
import java.io.*;
import java.util.Scanner;
public class Htod{
    public static void main(String []args)
    {
        Scanner in = new Scanner(System.in);
        String hex = in.nextLine();
        long dec = Long.parseLong(hex, 16);
        System.out.println(dec);
        in.close();
    }
}
```

■ STFW

# 题目讲解

*you want it, you print it.*

编写程序 Print.java, 它有以下命令行选项, 根据不同的选项得到不同的运行结果.

选项	用法举例	说明
-t	<code>`java Print -t type`</code>	若 type=n 则输出0到9的数字, type=a 则输出a到z的字母, 默认 type=n (即不带 -t 选项执行 <code>`java Print`</code> 将输出数字)
-o	<code>`java Print -o out.txt`</code>	输出到文件out.txt. 默认输出到标准输出
-h	<code>`java Print -h`</code>	输出帮助信息到标准输出, 不输出其他信息

例如 ``java Print -t a -o a.txt`` 将输出 a 到 z 到文件 a.txt. ``java Print -o b.txt`` 输出数字0到9到 b.txt. ``java Print -t a``, 将输出 a 到 z 到标准输出.

## ■ 参数

# 📖 题目讲解

*you want it, you process it.*

首先划重点：

## 数字图像

一幅数字图像 (digital image) 在计算机内部被表示成**像素矩阵** (pixel matrix).

每一个像素都有自己的颜色. 类比来说, 每个像素好比拼图玩具中的一块, 它们单个看起来都只是简单的色块, 但不同的块组合在一起就能构成完整的图画.

所以：本质上我们只是在处理一堆数字堆成的矩阵而已。

所以：

给定一个图片文件, 以及参数  $w, h$ , 将其转换成为宽为  $w$ , 高为  $h$  的图片.

当我们在缩放图片的时候, 我们在缩放什么?

数据损失怎么办?

- 如何在放大之后能“还原”一部分? 超分辨率
- 矢量图



# 📄 题目讲解

*you want it, you grey it.*

下面我们将以计算颜色的亮度 (Luminance) 为例, 说明如何使用 Color 类. 颜色的亮度表征了显示该颜色时需要使用多少强度的光. 直观上亮度越大, 该颜色越明亮. 严格的定义如下:

亮度 =  $0.299 * r + 0.587 * g + 0.114 * b$

其中,  $r, g, b$  分别代表该颜色的红, 绿, 蓝强度.

给定一个图片文件, 将其转换成为相应的灰度图 (Grayscale). 当 RGB 颜色模型中红绿蓝三色强度取相同值  $v$  时, 该颜色为灰色, 其灰度值为  $v$  ( $0 \leq v \leq 255$ ).

图片的灰度图可以通过如下方法得到: 将图中每个像素点的颜色置为灰色, 且这些灰色的灰度值等于该颜色亮度.

给定一个图片文件, 以及参数  $N$ . 输出  $N$  张图片, 其中第  $n$  张图片为从输入图片和其对应灰度图的一个渐变图.

- 灰色定义已给;
- 渐变? 映射。

# 📖 题目讲解

*you want it, you convolute it.*

滤镜 (filters) 可以看作对像素矩阵的某种变换. 通过添加滤镜, 我们可以改变图片的视觉效果. 请实现以下滤镜.

- Linear filter: 每一个像素的颜色变为周围 9 个像素点(包含它本身)颜色的平均值. 这个变换等价于将矩阵 
$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$
 与 9 个像素点做 卷积 (Convolution). 而这个矩阵被称为卷积矩阵 (convolutional matrix). 关于卷积矩阵可以参考 [1](#), [2](#).

```
static Color convolution(Color [][]a, double [][]c,int cw,int ch){
    int i,j;
    double R=0.0d,G=0.0d,B=0.0d;
    for(i=0;i<3;i++) for(j=0;j<3;j++){
        if(i+cw-1<0 || i+cw-1>=a.length || j+ch-1<0 || j+ch-1>=a[0].length) continue;
        else
        {
            R=R+c[2-i][2-j]*a[i+cw-1][j+ch-1].getRed();
            G=G+c[2-i][2-j]*a[i+cw-1][j+ch-1].getGreen();
            B=B+c[2-i][2-j]*a[i+cw-1][j+ch-1].getBlue();
        }
    }
    Color temp = new Color(((int)R>255)?255:((R<0)?0:(int)R),((int)G>255)?255:((G<0)?0:(int)G),((int)B>255)?255:((B<0)?0:(int)B));
    return temp;
}
```

# </> 杂项

- 调试
- 注意事项

# </> 杂项

## ■ 调试

- 手动调试: ``System.out.println()``
- IDE调试: 打断点
- 其它 (Vscode): 装拓展, 打断点, 写json。

## ■ 注意事项

- 什么是字面意思
- 压缩包的命名规范: 实验编号\_学号\_姓名.zip (例如: Lab1\_10215102000\_陈东.zip)
- 建议使用Markdown