

我的struts分页算法的实现 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E6_88_91_E7_9A_84stru_c104_145676.htm 说到分页算法，一般WEB开发都会用到，我只是在我的实现技术上用了struts框架，其实原理都一样的。看了网上相当多的分页算法，有对的也有好多是错的，更有好多是不太优化的。还有以前自己在augmentum做的一个分页算法，总结了一些不足。决定重新再写一个分页算法。首先，应该写个bean来记录存储一些页面的属性分页大致需要如下属性：

```
private int currentPage = 1. // 当前页
private int totalPages = 0. // 总页数
private int pageRecorders = 5. // 每页5条数据
private int totalRows = 0. // 总数据数
private int pageStartRow = 0. // 每页的起始数
private int pageEndRow = 0. // 每页显示数据的终止数
private boolean hasNextPage = false. // 是否有下一页
private boolean hasPreviousPage = false. // 是否有前一页
private int nextPage = 0. // 下一页的页码
private int previousPage = 0. // 上一页的页码
```

然后这些属性之间是有联系的，我们可以在构造函数的时候就初始化一些属性有两种方法：一，根据总的页数，（假设当前页为1）

```
public PageBean(int totalRows){
    this.totalRows = totalRows;
    this.currentPage = 1;
    hasPreviousPage = false;
    if ((totalRows % pageRecorders) == 0) {
        totalPages = totalRows / pageRecorders;
    } else {
        totalPages = totalRows / pageRecorders + 1;
    }
    if (totalRows >= pageRecorders) {
        hasNextPage = true;
        nextPage = 2;
        this.pageEndRow = pageRecorders;
    } else {
        this.pageEndRow = totalRows;
        hasNextPage = false;
        nextPage = 1;
    }
    this.pageStartRow =
```

0. previousPage = 1. } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com