

重 庆 大 学

学 生 实 验 报 告

实验课程名称 JAVA EE 程序设计

开课实验室 DS1501

学 院 软件学院 年级 2023 专业班 软件工程 01 班

学 生 姓 名 学 号

开 课 时 间 2024 至 2025 学年第 2 学期

成 绩	
教师签名	

大数据与软件学院制

《JAVA EE 程序设计》实验报告

开课实验室：DS1501

2025 年 3 月 16 日

学院	大数据与软件学 院	年级、专业、班	20 级软件 01 班	姓名		成绩	
课程 名称	JAVA EE 程序设计		实验项目 名 称	实验5-6：代码调试及数据 结构实现		指导教师	鄢萌
教 师 评 语							
<p>一、实验目的</p> <p>实验五：实现理论课上讲述的内容 PPT 内容的验证。</p> <p>实验六：掌握利用JAVA完成数据结构课程中的内容。</p> <p>选择：链表、栈、队列或二叉树等典型的数据结构其中的一个，实现其代码封装。实现该数据结构的基本方法：提取元素、插入元素、删除一个元素、找匹配某元素的个数等。具体内容参考《数据结构》课程内容介绍（在实验报告中增加该部分内容）。</p> <p>二、实验内容</p> <p>实验五内容：</p> <p>调试，验证，课件ppt内容，课后布置的作业（实验报告内容中可不写）。</p> <p>完成教材关于抽象类与接口方面的范例内容的代码验证、后面作业的代码设计（实验报告中可不写）。</p> <p>实验六内容：</p> <p>选择：链表、栈、队列或二叉树等典型的数据结构其中的一个，实现其代码封装。实现该数据结构的基本方法：提取元素、插入元素、删除一个元素、找匹配某元素的个数等。具体内容参考《数据结构》课程内容介绍（在实验报告中增加该部分内容）。</p> <p>三、使用仪器、材料</p> <p>JAVA SE JDK 1.8.121</p>							

四、实验过程原始记录(数据、图表、计算等):

```
1 package stack;
2 //pop,push,delete,match_number
3 public class Main {
4
5     public static void main(String[] args) {
6         // TODO 自动生成的方法存根
7         StackArray as=new StackArray(10);
8         as.push("nihao");
9         as.push(123);
10        as.delete();
11        as.pop();
12        for(int i=0;i<10;i++) {
13            as.push(i);
14        }
15        as.show();
16        as.remainder();
17    }
18 }
19
20 }
21
```

弹出栈顶nihao
0 1 2 3 4 5 6 7 8 9 剩余空间: 0

```
1 package stack;
2
3 public class StackArray<T> {
4     Object array[];
5     int top;
6     int upper;
7     public StackArray(int upper) {
8         // TODO 自动生成的构造函数存根
9         if(upper<=0) {
10             System.out.println("申请空间错误,初始空间需大于零");
11         }
12         else {
13             this.upper=upper;
14             array=new Object[upper];
15             top=0;
16         }
17     }
18     void push(T element) {
19         if(top<10) {
20             array[top]=element;
21             top++;
22         }
23         else {
24             System.out.println("栈满");
25         }
26     }
27
28     void pop() {
29         if(top!=0) {
30             System.out.println("弹出栈顶"+array[top-1]);
31             top--;
32         }
33         else {
34             System.out.println("栈空");
35         }
36     }
37 }
```

```

24     else {
25         System.out.println("栈满");
26     }
27 }
28* void pop() {
29     if(top!=0) {
30         System.out.println("弹出栈顶"+array[top-1]);
31         top--;
32     }
33     else {
34         System.out.println("栈空");
35     }
36 }
37* void delete() {
38     if(top!=0) {
39         top--;
40     }
41     else {
42         System.out.println("栈空");
43     }
44 }
45* void show() {
46     for(int i=0;i<top;i++)
47     {
48         System.out.print(array[i]+" ");
49     }
50 }
51* void remainder() {
52     if(top==9) System.out.println("栈满");
53     else if(top==0) System.out.println("栈空");
54     else {
55         System.out.println("剩余空间: "+(10-top));
56     }
57 }

```

实现了任意数据类型的入栈，出栈，删除，显示剩余和展示整个栈