**常用实用类**

**实 验 报 告**

**学 院** 信息科学与技术学院

**专 业** 计算机科学与技术

**班级/学号**  b22011108

**学生 姓名**  陈龙

**成 绩**

**成绩评定评语：**

**实 验 地 点**

**实 验 日 期 2022.11.03\_**

**指 导 教 师 马**永恒

上海杉达学院

电子商务

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 常用实用类 | 实验地点 | 线上 | 实验时间 | 2022.11.03 |
| 1. 实验目的：   熟练掌握常用实用类的编码方法； | | | | | |
| 1. 实验内容：     **教材p244：五、编程题** | | | | | |
| 1. 实验要求：   截屏  提供关键代码  a.在钉钉提交作业  b.源码放附在报告后  c.实验报告为word文档  d.上交截止时间为11.8日23:59:59 | | | | | |
| 1. 实验准备：  * 常用类概念知识和实现； * 实用类概念知识和实现； * Dev\_Tools:IDEA * JDK\_Ver:14.0.1 | | | | | |
| 1. 实验过程：（含实验具体内容、时间、实验过程描述及截图、遇到的问题截图及描述、解决方法等，如果有代码则需要附**代码**）：   （1）教材p224：五、编程题  **实验时间：**  2022年11月03日 星期四  **实验内容及过程：**   * 分析题目，对程序功能分析，完成函数功能，完成程序功能。 * 题目1：能将一串大小写混合的字符串完成大小写转换，如：AeBfC -> aEbFc。 * 题目2：获取一串字符串的首元素和尾元素，如：String -> S 和 g。 * 题目3：获取一段日期的间隔，如 2001/11/10-2022/11/03 -> 20年、239月、7298日。 * 题目4：Math类的使用，使用Math类的方法，如：abs, pow, sqrt, ciel, floor, max, min, sin, asin等。 * 题目5：通过正则表达式，将字符串中的英文字母去除，仅保留数字。 * 题目6：通过正则表达式，提取数字，并求出数字平均分。   **遇到的问题和解决方法：**  **问题：**   * 完成题目1时，使用了Atchar()方法，获取到的为char，与String.concat()，数据类型不符合；   解决方法：   * String.concat(String.valueOf(Character.toLowerCase(c)))   **代码（核心代码即可）：**   * A(源码见附页)：      * B：      * C：      * D：      * E：      * F：     **收获和体会：**  功能的实现就如同搭积木一般，每个看似复杂的功能，实际上抽丝剥茧下来，都是基础函数功能的叠加从而实现的。常用的字符串拼接除了常见的 + 拼接，还能通过concat()方法来完成。对于字符串中每个元素的获取亦可以通过首元素+相对位置来获取到。  对于时间的处理，相对其他而言，基础较薄弱，但是通过本次实验收获颇丰。可以通过格式化format输入和String.split()方法分割，学习颇丰，收获满满。 | | | | | |
| 1. 实验总结（总结该实验的总体体会和收获，还可以对这部分内容的教学方法提出建议）：   在第八章的学习中，主要学习了常用类和实用类，确实如其名，十分常用和实用，基础性的内容往往才是最为重要的。  今天学到的印象最深的就是hashcode，和C语言的指针不同，hashcode更偏向一个离散量，通过这个离散量能判断是否为一个类的实例或子类，在override方法时，也一定要重写hashcode，以防止后续程序出现难以察觉的错误。  正则表达式更是重要，无论是字符串匹配还是数据提取，都需要用到正则表达式。  Math类也有常见的数学表达式，减少不必要的代码，可以直接调用。  本次学习收获颇丰，逐步深入Java学习。 | | | | | |

说明：

1. **实验名称、实验目的、实验内容、实验要求**由教师确定，实验前由教师事先填好，然后作为实验报告模版供学生使用；
2. **实验准备**由学生在实验或上机之前填写，教师应该在实验前检查；
3. **实验过程**由学生记录实验的过程，包括操作过程、遇到哪些问题以及如何解决等；
4. **实验总结**由学生在实验后填写，总结本次实验的收获、未解决的问题以及体会和建议等；
5. 源程序、代码、具体语句等，若表格空间不足时可作为**附录**另外附页。

**附页：**

源码如下：

**A\_Application:**

1. **import** java.util.Scanner;
3. **public** **class** A\_Application {
4. **public** **static** **void** main (String[] ags){
5. String str = **null**;
6. Scanner scanner = **new** Scanner(System.in);
7. str = scanner.nextLine();
8. System.out.println("The Toggle of "+str+" is :" + ToggleCase(str,str.length()));
9. }
11. **static** String ToggleCase(String str,**int** n){
12. String ans = **new** String();
13. **for** (**int** i = 0; i < str.length(); i++){
14. **char** c = str.charAt(i);
15. **if**(Character.isLowerCase(c)){
16. //Change into UpperCase then add by concat() with ans
17. ans = ans.concat(String.valueOf(Character.toUpperCase(c)));
18. }**else**{
19. //Change into LowerCase then add by concat() with ans
20. ans = ans.concat(String.valueOf(Character.toLowerCase(c)));
21. }
22. }
23. **return** ans;
24. }
25. }

**B\_Application:**

1. **import** java.util.Scanner;
3. **public** **class** B\_Application {
4. **public** **static** **void** main (String[] ags){
5. String str = **null**;
6. Scanner scanner = **new** Scanner(System.in);
7. str = scanner.nextLine();
8. **char** first = str.charAt(0);                 //Get the first character
9. **char** last = str.charAt(str.length() - 1);   //Get the last character
10. System.out.println("The first character is :"+first+" .");
11. System.out.println("The Last character is :"+last+" .");
12. }
13. }

**C\_Application:**

1. **import** java.time.LocalDateTime;
2. **import** java.time.temporal.ChronoUnit;
4. **public** **class** C\_Application {
5. **public** **static** **void** main (String[] args){
6. **int**[] startDate = **new** **int**[5];
7. **int**[] endDate = **new** **int**[5];
8. **long** year, month, day;
9. **for**(**int** i = 0; i < 5; i++){
10. startDate[i] = Integer.parseInt(args[i]);    //Get Start date form main args
11. }
12. **for**(**int** i = 5; i < 10; i++){
13. endDate[i-5] = Integer.parseInt(args[i]);   //Get End date from main args
14. }
15. LocalDateTime dateStart = LocalDateTime.of(startDate[0],startDate[1],
16. startDate[2],startDate[3],startDate[4]);
17. LocalDateTime dateEnd = LocalDateTime.of(endDate[0],endDate[1],endDate[2],
18. endDate[3],endDate[4]);
19. year = dateStart.until(dateEnd,ChronoUnit.YEARS);
20. month = dateStart.until(dateEnd,ChronoUnit.MONTHS);
21. day = dateStart.until(dateEnd,ChronoUnit.DAYS);
22. System.out.println(args[0]+"-"+args[1]+"-"+args[2] + "  和 "+ args[5]+"-"+
23. args[6]+"-"+args[7] +"相差：");
24. System.out.println(year + "年");
25. System.out.println(month + "月");
26. System.out.println(day + "日");
27. }
28. }

**D\_Application:**

1. **import** **static** java.lang.Math.\*;
3. **public** **class** D\_Application {
4. **public** **static** **void** main(String[] args){
5. System.out.println("-12的绝对值为 ："+abs(-12.00));
6. System.out.println("98.89 和 6498.00的最大值为 ："+ max(98.89, 6498.00));
7. System.out.println("98.89 和 6498.00的最小值为 ："+ min(98.89, 6498.00));
8. System.out.println("0-1之间的随机数 ：" + random());
9. System.out.println("2的10次方为 ：" + pow(2,10));
10. System.out.println("36的算数平方根为 ：" + sqrt(36));
11. System.out.println("8的对数为 ：" + log(8));
12. System.out.println("45的正弦值为 ：" + sin(45));
13. System.out.println("45的反正弦值为 ：" + asin(45));
14. System.out.println("-12.56四舍五入为 ："+ round(-12.56));
15. System.out.println("12.56四舍五入为 ："+ round(12.56));
16. }
17. }

**E\_Application:**

1. **import** java.util.Scanner;
3. **public** **class** E\_Application {
4. **public** **static** **void** main(String[] args) {
5. String str = **null**;
6. Scanner scanner = **new** Scanner(System.in);
7. str = scanner.nextLine();
8. String regex = "[a-zA-z]";
9. System.out.println(str + "去掉字母为：" + str.replaceAll(regex,""));
10. }
11. }

**F\_Application:**

1. **import** java.util.InputMismatchException;
2. **import** java.util.Scanner;
4. **public** **class** F\_Application {
5. **public** **static** **void** main(String[] args){
6. String str = **null**;
7. Scanner scanner = **new** Scanner(System.in);
8. str = scanner.nextLine();
9. System.out.println("The average score is : " + getAvgScore(str));
10. }
11. **public** **static** **double** getAvgScore(String str){
12. Scanner scanner = **new** Scanner(str);
13. scanner.useDelimiter("[^0123456789.]+"); //only number accepted by the regex expression
14. **double** sum = 0;
15. **int** count = 0;
16. **while** (scanner.hasNext()){
17. **try**{
18. **double** score = scanner.nextDouble();
19. sum += score;
20. ++count;
21. }**catch**(InputMismatchException se){
22. String t = scanner.next();
23. }
24. }
25. **return** (sum / count);
26. }
27. }