【实验7-3】 日记本

### 【任务介绍】

**1.任务描述**

编写一个日记本功能的程序，使用字节流经日记的具体信息记录在本地的txt文件中。当用户输入日记的特定内容后，会将输入的内容保存至本地的txt文件中。需要输入的内容包括“姓名”，“天气”、“标题”、“内容”的数据。保存的时候需要判断本地是否存在文件，如果存在则追加，不存在则新建。文件命名格式为“黑马日记本”加上“.txt”后缀，如“黑马日记本.txt”

**2.运行结果**

任务运行结果如图7-1所示。

手机屏幕截图

描述已自动生成

图7-2 运行结果

运行结束后在本地生成一个“黑马日记本.txt”文件，如图7-2所示。

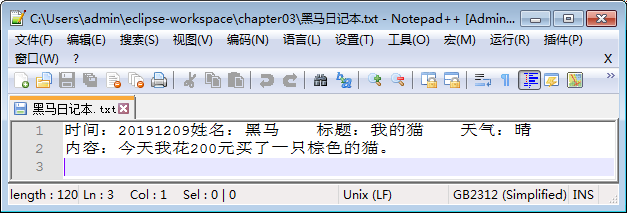


图7-2 黑马日记本.txt

### 【任务目标】

* 学会分析"日记本小功能"程序的实现思路。
* 根据思路独立完成"日记本小功能"的源代码编写、编译及运行。
* 掌握字节流操作本地文件的方法。
* 掌握ArrayList和StringBuffer的使用，以及异常的处理。

### 【实现思路】

（1）为方便保存日记的相关信息，可以将日记信息封装成一个实体类。

（2）用户编写日记时，首先创建一个集合用于存放日记，然后用户依次填写的内容为“姓名”，“天气”、“标题”和“内容”，并将这些内容存放在集合中，再将实体类保存到txt文件中。

（3）查询日记时，使用字节流的读取实现控制台打印日记信息。

（4）将日记信息写入到txt文件之前，需先拼凑好txt文件名，再判断本地是否已存在此文件，这里可通过输入流尝试获取此文件的字节流，如果获取成功，则证明这个文件已存在，那么就通过输出流向文件末尾追加日记信息，如果获取失败，即异常，说明之前并没有生成日记信息，则需要新建此文件。

### 【实现代码】

日记小功能的实现代码，如文件7-1所示。

（1）将日记信息封装成一个实体类Diary，具体如文件7-1所示。

文件7-1 Diary.java

1. package chapter0702;
2. public class Diary {
3. String time; //时间
4. String name; //姓名
5. String weather;//天气
6. String title;//标题
7. String content;//内容
8. public Diary(String time, String name, String weather, String title,
9. String content) {
10. this.time = time;
11. this.name = name;
12. this.weather = weather;
13. this.title = title;
14. this.content=content;
15. }
16. }

在文件7-1中，第3-7行代码定义了用于标识日记信息的各个字段，第8-14行代码重新了toString()方法。

（2）定义DiaryOrder类来记录和操作日记信息，具体如文件7-2所示。

文件7-2 DiaryOrder.java

1. package chapter0702;
2. import java.text.DateFormat;
3. import java.text.SimpleDateFormat;
4. import java.util.ArrayList;
5. import java.util.Date;
6. import java.util.Scanner;
7. public class DiaryOrder {
8. static ArrayList<Diary> diarysList=new ArrayList<Diary>();
9. public static void main(String[] args) throws Exception {
10. System.out.println("--------欢迎来到黑马日记本--------");
11. boolean falg = true;
12. Scanner scan=new Scanner(System.in);
13. while(falg) {
14. System.out.println("1.编写日记");
15. System.out.println("2.查看日记");
16. int a = scan.nextInt();
17. if(a==1) {
18. //编写日记
19. System.out.println("请输入姓名：");
20. String name=scan.next();
21. System.out.println("请输入天气：");
22. String weather=scan.next();
23. System.out.println("请输入标题：");
24. String title=scan.next();
25. System.out.println("请输入内容：");
26. String content=scan.next();
27. Diary diary = addDiary(name,weather,title,content);
28. FileUtil.saveBooks(diary);
29. }else if(a==2) {
30. //查看日记
31. FileUtil.readFile();
32. }else {
33. System.out.println("输入错误");
34. }
35. }
36. }
37. /\*
38. \* 初始化日记列表 将输入的字符暂存为实体
39. \*/
40. private static Diary addDiary(String name,String weather,String
41. title,String content) {
42. Date date = new Date();
43. DateFormat format = new SimpleDateFormat("yyyyMMdd");
44. String a=format.format(date).toString();
45. Diary diary=new Diary(a,name,weather,title,content);
46. return diary;
47. }
48. }

在文件7-2中，第8行代码创建了ArrayList类型的全局变量作为存储日记的仓库。第13~35行代码使用while循环输入操作，第19~26行代码让用户输入日记，第27~28行代码将用户输入的日记存入本地文件，第31行代码调用readFile（）方法显示日记，第40~47行代码初始化日记信息。

（3）定义工具类FileUtil保存日记信息，具体如文件7-3所示。

文件7-3 FileUtil.java

1. package chapter0702;
2. import java.io.BufferedOutputStream;
3. import java.io.FileInputStream;
4. import java.io.FileNotFoundException;
5. import java.io.FileOutputStream;
6. import java.io.IOException;
7. import java.io.InputStream;
8. import java.text.DateFormat;
9. import java.text.SimpleDateFormat;
10. import java.util.Date;
11. /\*\*
12. \* 工具类
13. \*/
14. public class FileUtil {
15. public static final String SEPARATE\_FIELD = "\n";// 换行
16. public static final String SEPARATE\_LINE = "\t"; // 分隔
17. /\*\*
18. \* 保存日记信息
19. \*/
20. public static void saveBooks(Diary diary) {
21. // 判断本地是否存在此文件
22. String name = "黑马日记本.txt";
23. InputStream in = null;
24. try {
25. in = new FileInputStream(name);// 判断本地是否存在此文件
26. if (in != null) {
27. in.close();// 关闭输入流
28. // 可获取输入流，则存在文件，采取修改文件方式
29. createFile(name, true, diary);
30. }
31. } catch (FileNotFoundException e) {
32. // 输入流获取失败，则不存在文件，采取新建新文件方式
33. createFile(name, false, diary);
34. } catch (IOException e) {
35. e.printStackTrace();
36. }
37. }
38. /\*\*
39. \* 将日记信息保存到本地，可通过label标识来判断是修改文件还是新建文件
40. \* @param name 文件名
41. \* @param label 文件已存在的标识 true：已存在则修改； false：不存在则新建
42. \* @param diary 日记信息
43. \*/
44. public static void createFile(String name, boolean label, Diary diary) {
45. BufferedOutputStream out = null;
46. StringBuffer sbf = new StringBuffer();// 拼接内容
47. try {
48. if (label) {// 当已存在当天的文件，则在文件内容后追加
49. // 创建输出流，用于追加文件
50. out = new BufferedOutputStream(new FileOutputStream(name, true));
51. } else {// 不存在当天文件，则新建文件
52. // 创建输出流，用于保存文件
53. out = new BufferedOutputStream(new FileOutputStream(name));
54. String fieldSort = "欢迎来到黑马日记本" ;// 创建表头
55. // 新建时，将表头存入本地文件
56. sbf.append(fieldSort).append(SEPARATE\_FIELD);
57. }
58. sbf.append("时间：").append(diary.time);// 追加换行符号
59. sbf.append("姓名：").append(diary.name).append(SEPARATE\_LINE);
60. sbf.append("标题：").append(diary.title).append(SEPARATE\_LINE);
61. sbf.append("天气：").append(diary.weather).append(SEPARATE\_FIELD);
62. sbf.append("内容：").append(diary.content).append(SEPARATE\_FIELD);
63. String str = sbf.toString();
64. byte[] b = str.getBytes();
65. for (int i = 0; i < b.length; i++) {
66. out.write(b[i]);// 将内容写入本地文件
67. }
68. } catch (Exception e) {
69. e.printStackTrace();
70. } finally {
71. try {
72. if (out != null)
73. out.close();// 关闭输出流
74. } catch (Exception e2) {
75. e2.printStackTrace();
76. }
77. }
78. }
79. /\*
80. \* 读取日记并显示
81. \*/
82. public static void readFile() throws Exception{
83. //创建文件字节输出流
84. FileInputStream in = new FileInputStream("黑马日记本.txt");
85. byte[] b = new byte[in.available()];
86. in.read(b);
87. String str=new String(b);
88. System.out.println(str);
89. in.close();
90. }
91. }

在文件7-3中，当saveBooks()方法被调用时，获取当前日期并格式化后，拼出了文件名，再通过第29行代码尝试获取此文件的字节输入流。当能够获取输入流时，可通过第27~29行代码，先关闭输入流，再在文件末尾追加信息。当不能获取输入流时则抛出异常，在异常处理中调用第33行代码的createFile()方法，可以通过此方法中的label参数来区分，是新建文件还是在已有文件中追加内容，如果label值是true则追加内容，如果label值是false则新建文件，并写入表头，其中进行追加还是新建操作，由构造函数的append参数来定义。然后利用58~63行代码拼出一行数据，且在每次拼接之前都要加上换行符“\r\n”，每个字段之间通过“,”分隔字段，再利用第64~67行的代码写入文件。最后71~77行代码关闭了输出流。