

자바프로그래밍응용 보고서

과제 1 : 전화번호부 프로그램



강의명	자바프로그래밍응용 002 분반
담당교수	정문주
학과	소프트웨어학과
학번	2023864041
이름	정지민
제출일자	2024.09.29

목차

표 지	1
목 차	2
프 로 그 램 목 표	2
코 드 구 현	3
public class Phone	3
main Method	3
WritePhoneNumber Method	3
ReadPhoneNumber Method	4
SearchPhoneNumber Method	4
이하 추가 구현 Method	5
코 드 실 행	6
프로그램 실행	6
입력	6
출력	6
검색	7
파일경로변경	7
종료	7

프로그램 목표

교재의 8 장 실습문제 1 번, 2 번, 10 번 구현하기.

1 번 : Scanner 로 한 줄에 한 명씩 이름과 전화번호를 입력 받아 텍스트 파일로 저장하기.

2 번 : 앞서 저장한 텍스트 파일을 읽어 화면에 전체 전화번호부 출력하기.

10 번 : 전화번호 검색 프로그램. 이름을 입력하면 전화번호를 찾아 출력.

*JDK 23 버전(2024 일 9 월 17 일 릴리스)를 사용하여 코드를 작성하였습니다.

코드 구현

1. 클래스 파일 Phone

```
public class Phone {  
    static Scanner in = new Scanner(System.in); //스캐너 클래스 변수로 선언  
    static File file = new File("c:\\Temp\\Phone.txt"); //기본 파일 경로 선언
```

static 클래스 변수로 키보드 입력을 받는 스캐너와 기본 파일 경로를 선언합니다.

2. main Method

```
10● public static void main(String[] args) {  
11     PrintStrings();  
12     while(in.hasNext()) {  
13         String command = in.next(); in.nextLine();  
14         switch (command) {  
15             case "입력": { WritePhoneNumber(file); PrintStrings(); break; }  
16             case "출력": { ReadPhoneNumber(file); PrintStrings(); break; }  
17             case "검색": { SearchPhoneNumber(file); PrintStrings(); break; }  
18             case "파일경로변경" : { getFileName(); PrintStrings(); break; }  
19             case "종료": {  
20                 System.out.println("프로그램을 종료합니다.");  
21                 in.close();  
22                 System.exit(0);  
23             }  
24             default:  
25                 System.out.println("잘못된 입력입니다. 다시 입력해주세요.");  
26             }  
27         }  
28     }
```

프로그램의 메인 구조입니다. Scanner 를 통해 문자열 `command` 를 입력받고, case 문에 따라 명령어에 따라 각 메서드를 실행합니다. 개행 문자로 인한 입력 오류를 없애기 위해 `nextLine()`을 사용해 버퍼를 비웁니다. "종료"를 입력받으면 Scanner 를 닫고 프로그램을 종료합니다.

3. WritePhoneNumber Method

```
30● public static void WritePhoneNumber(File f) { //실습문제 1. 전화번호 저장  
31     FileWriter fout = null;  
32     try {  
33         fout = new FileWriter(f);  
34         System.out.println("전화번호 입력 프로그램입니다.");  
35         System.out.println("입력 종료를 원할 시 \'그만\'을 입력해주세요.");  
36         while(true) {  
37             System.out.print("이름 전화번호 >> ");  
38             String line = in.nextLine();  
39             if(line.equals("그만")) break;  
40             if(isCorrect(line)) { //입력 정보 판별 후 저장  
41                 fout.write(line, 0, line.length());  
42                 fout.write("\r\n", 0, 2);  
43             }  
44             else {  
45                 System.out.println("올바른 입력 형식이 아닙니다.");  
46             }  
47         }  
48         System.out.println(f.getPath()+"에 저장하였습니다.");  
49         fout.close();  
50     } catch (IOException e) {  
51         System.out.println("작성에 실패하였습니다. 경로를 확인해보세요.");  
52     }  
53 }
```

FileWriter 출력 스트림을 생성하고 출력할(작성할) 파일 `f` 를 연결합니다. Scanner 를 통해 문자열 `line` 을 입력받습니다. 올바른 이름, 전화번호의 형식인지 판별 후 `line` 과 개행문자(줄 띄어쓰기)을 파일에 작성합니다.

4. ReadPhoneNumber Method

```
55● public static void ReadPhoneNumber(File f) { //실습문제 2. 전화번호 출력
56     FileReader fin = null;
57     int c;
58     try {
59         fin = new FileReader(f);
60         System.out.println(f.getPath()+"를 출력합니다.");
61         while((c = fin.read()) != -1) {
62             System.out.print((char)c);
63         }
64         fin.close();
65     } catch (IOException e) {
66         System.out.println("출력에 실패하였습니다. 경로를 확인해보세요.");
67     }
68 }
69 }
```

FileReader 입력 스트림을 생성하고 파일 *f* 를 연결합니다. 파일 내 자료가 없을 때까지 *.read()*를 사용해 int 자료형으로 문자를 하나씩 읽습니다. char 자료형으로 캐스팅하여 출력합니다.

5. SearchPhoneNumber Method

```
71● public static void SearchPhoneNumber(File f) { //실습문제 10. 전화번호 검색
72     HashMap<String, String> nums = new HashMap<>();
73     int count = 0;
74     try {
75         BufferedReader bin = new BufferedReader(new FileReader(f));
76         String line;
77         while((line = bin.readLine()) != null) {
78             String[] data = line.split(" ");
79             nums.put(data[0], data[1]);
80             count++;
81         }
82         System.out.printf("총 %d개의 전화번호를 읽었습니다.\n", count);
83         bin.close();
84
85         System.out.println("검색 종료를 원할 시 \'그만\'을 입력해주세요.");
86         while(true) {
87             System.out.print("이름 검색 >> ");
88             String text = in.nextLine();
89
90             if(text.equals("그만")) break;
91
92             if(nums.containsKey(text)) {
93                 System.out.println(nums.get(text));
94             }
95             else {
96                 System.out.println("찾는 이름이 없습니다.");
97             }
98         }
99     } catch (IOException e) {
100         System.out.println("파일을 읽어오지 못했습니다. 경로를 확인해보세요.");
101     }
102 }
103 }
```

이름과 전화번호를 저장할 <String, String> 형태의 HashMap 을 선언합니다. FileReader 로 파일 *f* 와 연결한 후 한 줄씩 읽기 위해 BufferdReader 를 생성합니다. *.readline()*을 사용해 더 이상 읽을 데이터가 없을 때까지 이름은 key 값에, 전화번호는 value 값에 저장합니다. Scanner 를 이용하여 검색할 이름을 입력 받고 Hashmap 의 key 값을 이용해 전화번호를 탐색합니다.

6. 이하 추가 구현 Method

```
104● private static void getFileName() { //파일 경로 입력 메서드
105     String File_name = null;
106     while(true) {
107         System.out.print("경로를 입력해 주세요 >> ");
108         File_name = in.next(); in.nextLine();
109         file = new File(File_name);
110         if(file.exists()) break;
111         else System.out.println("파일을 찾을 수 없습니다.");
112     }
113 }
114
115● private static void PrintStrings() { //문구 출력 간편화
116     System.out.println("==== 전화번호부 프로그램 =====");
117     System.out.println("현재 파일 경로 : "+file.getPath());
118     System.out.println("\'입력\' \'출력\' \'검색\' \'파일경로변경\' 중 원하는 기능을 입력해주세요.");
119     System.out.println("\'종료\' 입력 시 프로그램이 종료됩니다.");
120 }
121
122● private static boolean isCorrect(String line) { //이름, 전화번호 형식 판별
123     String[] data = line.split(" ");
124
125     if(data.length != 2) return false;
126
127     String name = data[0];
128     String number = data[1];
129
130     if(name.length() > 0 && number.matches("\\d{3}-\\d{4}-\\d{4}")) return true;
131     return false;
132 }
133 }
```

getFileName : String 자료형의 파일명을 입력받아 File 클래스를 생성합니다.

PrintStrings : 전화번호부 프로그램의 기본 문구를 출력합니다.

isCorrect : 문자열을 매개변수로 받습니다. 정규표현식을 이용하여 이름이 0 자를 초과하고, 전화번호가 xxx-xxxx-xxxx 형태를 취하고 있는지 판별하여 true 또는 false 를 반환합니다.

코드 실행

1. 프로그램 실행

```
===== 전화번호부 프로그램 =====  
현재 파일 경로 : c:\Temp\Phone.txt  
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.  
'종료' 입력 시 프로그램이 종료됩니다.  
예외처리테스트  
잘못된 입력입니다. 다시 입력해주세요.  
|
```

기본 문구가 출력됩니다. 키워드와 다른 단어를 입력할 시 다시 입력을 받습니다.

2. 입력

```
입력  
전화번호 입력 프로그램입니다.  
입력 종료를 원할 시 '그만'을 입력해주세요.  
이름 전화번호 >> 황기태 010-5555-7777  
이름 전화번호 >> 이재문 011-3333-4444  
이름 전화번호 >> 김남윤 065-2222-1111  
이름 전화번호 >> 최박사 010-2222-3333  
이름 전화번호 >> 백점만 011-1100-1100  
이름 전화번호 >> 이상형 010-3333-4444  
이름 전화번호 >> 이름만  
올바른 입력 형식이 아닙니다.  
이름 전화번호 >> 010-0000-0000  
올바른 입력 형식이 아닙니다.  
이름 전화번호 >> 테스트 010-00-00  
올바른 입력 형식이 아닙니다.  
이름 전화번호 >> 그만  
c:\Temp\Phone.txt에 저장하였습니다.  
===== 전화번호부 프로그램 =====  
현재 파일 경로 : c:\Temp\Phone.txt  
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.  
'종료' 입력 시 프로그램이 종료됩니다.
```

3. 출력

```
출력  
c:\Temp\Phone.txt를 출력합니다.  
황기태 010-5555-7777  
이재문 011-3333-4444  
김남윤 065-2222-1111  
최박사 010-2222-3333  
백점만 011-1100-1100  
이상형 010-3333-4444  
===== 전화번호부 프로그램 =====  
현재 파일 경로 : c:\Temp\Phone.txt  
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.  
'종료' 입력 시 프로그램이 종료됩니다.
```

4. 검색

```
검색
총 6개의 전화번호를 읽었습니다.
검색 종료를 원할 시 '그만'을 입력해주세요.
이름 검색 >> 홍길동
찾는 이름이 없습니다.
이름 검색 >> 이상형
010-3333-4444
이름 검색 >> 그만
===== 전화번호부 프로그램 =====
현재 파일 경로 : c:\Temp\Phone.txt
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.
'종료' 입력 시 프로그램이 종료됩니다.
```

5. 파일경로변경

```
===== 전화번호부 프로그램 =====
현재 파일 경로 : c:\Temp\Phone.txt
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.
'종료' 입력 시 프로그램이 종료됩니다.
파일경로변경
경로를 입력해 주세요 >> 주소예외처리테스트
파일을 찾을 수 없습니다.
경로를 입력해 주세요 >> c:\Temp\Test.txt
===== 전화번호부 프로그램 =====
현재 파일 경로 : c:\Temp\Test.txt
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.
'종료' 입력 시 프로그램이 종료됩니다.
```

6. 종료

```
===== 전화번호부 프로그램 =====
현재 파일 경로 : c:\Temp\Test.txt
'입력' '출력' '검색' '파일경로변경' 중 원하는 기능을 입력해주세요.
'종료' 입력 시 프로그램이 종료됩니다.
종료
프로그램을 종료합니다.
```