

객체지향프로그래밍 - 15주차 실습 답안지

성명: 조우연 학과: 수학과 학번: 201921195

※ 본 실습활동지에 대한 보고서나 코드를 작성함에 있어 교재나 강의노트를 제외한 다른 자료로부터 일부 또는 전체를 복사하였습니까? 예() 아니오(O)

1. TextFileTest.java와 Employee.java를 각각 수행해 본 후 다음 물음에 답하시오.

(가) cmd 창에서 'employee.dat' 파일을 메모장을 통해 결과를 확인하시오.

- 수행결과

```
C:\Users\User\Desktop\객체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15
```

employee.dat - Windows 메모장

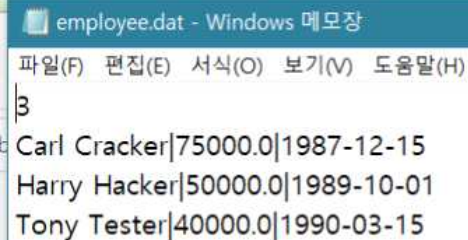
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15
```

(나) TextFileTest.java의 main함수에서 'UTF-8'을 모두 'UTF-16'으로 수정 후 실행한 후 결과를 비교하시오. 그리고 'employee.dat'를 메모장을 통해 결과를 확인하시오. (가)와 어떤 차이가 있는지 설명하고 왜 그런지 설명하시오.

- 수행결과

```
C:\Users\user\Desktop\객체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15
```



employee.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15

- 설명

main함수의 모든 "UTF-8"을 "UTF-16"으로 변경하면 입력할 문자들이 employee.dat 파일에 UTF-16으로 인코딩해 저장하게 된다. 마찬가지로 파일을 읽을 때에도 UTF-16으로 디코딩되어 읽게 된다. 따라서 UTF-16으로 파일을 작성하고, 파일을 읽을 때에도 UTF-16으로 파일을 읽는다면, 올바르게 파일을 작성하고 읽을 수 있게 된다. 또한 영어는 2바이트단위로 구성되어 있기 때문에, 2바이트씩 나누어 인코딩하는 UTF-16으로 바뀌어도 텍스트를 작성하고, 읽는데 문제가 없다. (다만 인코딩체계가 UTF-8에서 UTF-16으로 바뀌었으므로, cmd에서의 출력형태가 조금 바뀔 수 있는 것 같다.)

(다) TextFileTest.java의 main함수에서 쓸 때에는 encoding shcheme을 'UTF-8'을, 읽을 때에는 default scheme(encoding shcheme을 지정하지 않음)을 이용하여 수행해보고 반대의 경우에도 수행하여 그 결과를 비교하시오. 수행한 후 메모장을 통해 결과를 비교하시오.

- 수행결과 (작성 : UTF-8, 읽을 때 : default scheme)

```
C:\Users\user\Desktop\객체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15
```

employee.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

3

Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15

- 설명

UTF-8로 파일을 작성하고, UTF-8로 파일을 읽을 때와 같은 결과가 나왔다. 본인 컴퓨터의 default scheme을 확인해보니, x-windows-949 이었다. x-windows-949에 대해서 검색해보니 x-windows-949는 확장 완성형 인코딩체계로써, UTF-8에서 표현할 수 있는 모든 문자를 표현 가능하다는 것을 알 수 있었다. 따라서 파일 작성을 UTF-8으로, 파일읽기를 default scheme으로 하였을 때에는 파일을 안정적으로 작성하고 읽을 수 있다.

- 수행결과 (작성 : default scheme, 읽을 때 : UTF-8)

```
C:\Users\user\Desktop\객체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15
```

employee.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

3

Carl Cracker|75000.0|1987-12-15
Harry Hacker|50000.0|1989-10-01
Tony Tester|40000.0|1990-03-15

- 설명

위와 같은 결과가 나왔다. 마찬가지로 default scheme으로 설정되어 있는 x-windows-949가 확장 완성형 인코딩체계이므로 UTF-8에서 표현할 수 있는 모든 문자를 표현 가능하기 때문에 default scheme으로 파일을 작성하고, UTF-8로 파일을 읽는데 아무런 문제가 생기지 않는다.

(라) TextFileTest.java의 main함수에서 Employee의 name부분을 모두 한글로 수정한 후
(다)번처럼 수행하시오.

- 수행결과 (작성 : UTF-8, 읽을 때 : default scheme)

```
C:\Users\user\Desktop\책체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
議갇슌??75000.0|1987-12-15
議고웁??50000.0|1989-10-01
議갇웁??40000.0|1990-03-15
```

employee.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

3

조우연|75000.0|1987-12-15

조희연|50000.0|1989-10-01

조성연|40000.0|1990-03-15

- 설명

cmd에서는 한글텍스트가 깨지게 출력이 되었고, 메모장에서는 파일이 정상적으로 작성되어 저장되어있는 것을 관찰할 수 있다. 이러한 결과로 미루어 보아, 파일 작성은 정상적으로 처리되었지만, 파일을 읽는 과정에서 문제가 생겼음을 알 수 있다. 위에 작성한 것과 같이 default scheme으로 설정되어 있는 x-windows-949가 UTF-8로 표현할 수 있는 모든 문자를 표현가능하기 때문에, 정상적으로 파일을 읽을 수 있을 것이라는 예상과 다르게 cmd에서는 한글텍스트가 깨져서 출력되었다. 따라서 cmd의 인코딩체계가 windows의 인코딩체계와 다르다고 생각했는데, cmd의 인코딩체계를 확인해보니 cp949로 windows와 같은 인코딩체계를 가지고 있었다. cmd에서만 한글텍스트가 깨지는 이유를 알 수 없다.

- 수행결과 (작성 : default scheme, 읽을 때 : UTF-8)

```
C:\Users\user\Desktop\책체지향프로그래밍\lab_B_15_201921195>type employee.dat
3
조우연|75000.0|1987-12-15
조희연|50000.0|1989-10-01
조성연|40000.0|1990-03-15
```

employee.dat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

3

조우연|75000.0|1987-12-15

조희연|50000.0|1989-10-01

조성연|40000.0|1990-03-15

- 설명

Employee의 name 부분을 모두 한글로 수정한 후 실행하였을 때, 한글 텍스트가 깨지지 않고 파일이 정상적으로 작성되고 읽을 수 있다. x-windows -949가 확장 완성형 인코딩체계이므로 UTF-8에서 표현할 수 있는 모든 문자를 표현 가능하기 때문에 default scheme으로 파일을 작성하고, UTF-8로 파일을 읽는데 아무런 문제가 생기지 않는다.

(마) TextFileTest class에서 PrintWriter를 이용하여 파일에 쓰는 대신 StringWriter를 이용하여 String에 쓴 후 Scanner를 이용하여 읽어들이는 대신 StringWriter로부터 읽어들이고자 한다. 각 employee객체를 PrintWriter로 출력하여 StringWriter로부터 String값을 얻어 화면에 출력하도록 TextFileTest 코드를 수정하여 실행해보시오.

- 수정된 코드

```
public class TestFileTest {  
  
    public static void main(String[] args) throws IOException{  
        Employee[] staff = new Employee[3];  
        staff[0] = new Employee("Carl Cracker",75000,1987,12,15);  
        staff[1] = new Employee("Harry Hacker",50000,1989,10,1);  
        staff[2] = new Employee("Tony Tester",40000,1990,3,15);  
  
        StringWriter writer = new StringWriter();  
        try(PrintWriter out = new PrintWriter(writer))  
        {  
            writeData(staff,out);  
        }  
        System.out.println(writer.toString());  
    }  
}
```

-> TestFileTest의 main함수 코드만 수정되었음. 즉 캡처되지 않는 TestFileTest의 static method들은 기존 코드와 동일함.

- 수행결과

```
3  
Carl Cracker|75000.0|1987-12-15  
Harry Hacker|50000.0|1989-10-01  
Tony Tester|40000.0|1990-03-15
```


2. RandomAccess 프로그램을 이용하여 다음 물음에 답하시오.

(가) main함수의 맨 뒤에 다음 작업을 하도록 코드를 작성하여 삽입한 후, 앞에서부터 순서대로 employee를 출력하도록 수정하시오.

- 코드

```
public static void main(String[] args) throws IOException{
    Employee[] staff= new Employee[3];
    staff[0] = new Employee("Carl Cracker",75000,1987,12,15);
    staff[1] = new Employee("Harry Hacker",50000,1989,10,1);
    staff[2] = new Employee("Tony Tester",40000,1990,3,15);

    try(DataOutputStream out = new DataOutputStream(new FileOutputStream("employee.dat"))){
        for(Employee e : staff)
            writeData(out,e);
    }
    try(RandomAccessFile in = new RandomAccessFile("employee.dat","r")){
        int n = (int)(in.length()/Employee.RECORD_SIZE);
        Employee[] newStaff = new Employee[n];
        for(int i=n-1;i>=0;i--){
            in.seek(i*Employee.RECORD_SIZE);
            newStaff[i]=readData(in);
        }
        for(Employee e : newStaff)
            System.out.println(e);
    }

    System.out.printf("\n");

    //Harry Hacker의 연봉 10% 인상
    staff[1].raiseSalary(10);
    RandomAccessFile in2 = new RandomAccessFile("employee.dat","rw"); //읽어쓰기 위함
    //Harry Hacker의 위치 찾을
    in2.seek(1*Employee.RECORD_SIZE);
    writeData(in2,staff[1]); //변경된 연봉 반영하여 읽어쓰기

    //읽어들이기
    int n = (int)(in2.length()/Employee.RECORD_SIZE);
    Employee[] newStaff = new Employee[n];
    for(int i=n-1;i>=0;i--){
        in2.seek(i*Employee.RECORD_SIZE);
        newStaff[i]=readData(in2);
    }
    //인상된 연봉이 반영된 employee출력
    for(Employee e : newStaff)
        System.out.println(e);
    in2.close();
}
```

-> 형광펜 : main함수의 맨 뒤에 추가된 코드

- 수행결과

```
lab_B_15_201921195.Employee[name=Carl Cracker,salary=75000.0,hireDay=1987-12-15]
lab_B_15_201921195.Employee[name=Harry Hacker,salary=50000.0,hireDay=1989-10-01]
lab_B_15_201921195.Employee[name=Tony Tester,salary=40000.0,hireDay=1990-03-15]

lab_B_15_201921195.Employee[name=Carl Cracker,salary=75000.0,hireDay=1987-12-15]
lab_B_15_201921195.Employee[name=Harry Hacker,salary=55000.0,hireDay=1989-10-01]
lab_B_15_201921195.Employee[name=Tony Tester,salary=40000.0,hireDay=1990-03-15]
```

-> 형광펜 : Harry Hacker의 연봉을 10% 인상하는 작업을 반영하여 다시 출력한 결과

3. 소프트웨어학과에서 매학기 개최하는 종합작품전시회인 Ajou SOFTCON대회에 출품한 작품 중 3개를 감상한 후 감상평을 각각 작성하시오.

작품제목 : Deliversity

감상평 : 학교주변에 1인가구가 많다는 특성을 이용해 배달서비스앱을 개발한 아이디어가 매우 좋았다. 본인도 혼자 살았을 때, 음식 배달앱의 최소배달비용이 너무 비싸서 먹고 싶은 음식을 주문하지 못할 때가 많았는데, Deliversity를 이용하면 혼자서 주문하기 불편했던 음식을 주문할 수 있을 것 같다. 또한 음식점의 배달업체 이용이 증가하면서 배달비용이 심하게 비싸서 이용하기 어려울 때도 있었는데, 이러한 문제점을 Deliversity가 해결해줄 수 있을 것 같다. 앱개발이라는 것이 단순히 테마를 설정하고, 원하는 기능을 추가하는 것이라고 생각했는데, 회원정보저장의 방법, 다른 어플과의 연동 등 고려할 것이 상당히 많은 작업이라는 것을 알게 되었다.

작품제목 : 로켓폰트

감상평 : 실제로 스마트폰에 원하는 폰트를 다운받아서 사용하였을 때, 대부분의 앱이 느려지거나, 타자를 칠 때 렉을 먹는 등의 불편함이 발생할 때가 많았다. 그래서 폰트를 사용하지 않고, 스마트폰에 탑재된 기본 폰트를 사용하고는 했었다. 스마트폰을 사용하면서 폰트가 크게 중요한 부분은 아니지만, 누구든지 폰을 보기좋게 꾸미고자 한다면 폰트변경을 시도해보았을 것이라고 생각한다. 이러한 면에서 폰트변경 이후에도 앱이 느려지지 않는다는 것은 스마트폰 사용에 있어서 삶의 질이 높아지게 하는 획기적인 아이템이라고 생각된다. 또한 개발의 관점에서 봤을 때, 개발자는 적은 비용으로 사이트의 속도를 개선하여 수익을 증대시킬 수 있다는 면에서 일반적인 개발의 목적성을 잘 살린 작품인 것 같다.

작품제목 : 모두나눔

감상평 : “버리긴 아깝고, 두자니 잘 쓰지 않는 물건”들을 빌려주고 빌리자. 라는 아이디어는 매우 좋은 아이디어인 것 같다. 필요한 사람들은 물건을 빌릴 수 있고, 필요하지 않은 사람들은 작은 수익을 남길 수 있기 때문이다. 마찬가지로 앱을 통해 중고 대여서비스가 이루어지면 개발자도 수익을 남길 수 있기 때문에, 앱 개발자와 사용자에게 다다익선이 되는 좋은 앱인 것 같다. 앱 개발은 자체적으로 프로그램을 만드는 것이라고 생각했었는데, 개발 스택에 대한 내용을 보니, 카카오톡, 전자서명 프로그램, 백엔드 등 다양한 프로그램이 동원되어 사용되어 만들어진다는 것을 알게 되었다. 아직 프로그램이나, 웹, 서버 등에 대한 지식이 굉장히 부족하다는 것을 깨닫게 되었고, 앱개발에 사용된 다양한 프로그램의 종류들을 볼 수 있어서 굉장히 좋은 발표영상이었다고 생각한다.