

# 정리노트 #9

13조( 소재현, 전대원 )

**텍스트 파일:** 글자 혹은 문자들로만 구성되는 파일,

txt 파일, HTML 파일, XML 파일, C++ 소스 파일, C 소스 파일, 자바 소스 파일  
'\n', '\t' 등의 특수 문자도 포함

**바이너리 파일:** 문자로 표현되지 않는 바이너리 데이터가 기록된 파일

이미지, 오디오, 그래픽, 컴파일된 코드는 문자로 표현되지 않음

텍스트 파일의 각 바이트 -> 문자로 해석

바이너리 파일의 각 바이트 -> 문자로 해석되지 않는 것도 있음

각 바이트의 의미는 파일을 만든 응용프로그램 만이 해석 가능

ifstream : 파일 읽기 시에 사용

fstream : 하나의 파일에 대해 읽기와 쓰기를 동시에 할 때 사용

ofstream : 파일 쓰기 시에 사용

- #include <fstream>
- using namespace std; 코드에 써주어야함

## 예제 12-1

```
fout << sid << endl; // sid 쓰기 (정수 sid가 문자열로 변환되어 저장됨)
```

20131111 은 >>> 20131111이

'2', '0', '1', '3', '1', '1', '1', '1'의 문자들로 변환되어 저장

파일 모드 지정 – 파일 열 때

- open("파일이름", 파일모드)
- ifstream("파일이름", 파일모드),
- ofstream("파일이름", 파일모드)

## 예제 12-3

```
while((c=fin.get()) != EOF) { // EOF를 만날 때까지 문자 읽기
```

```
cout << (char)c;
```

```
count++;
```

```
}
```

파트에서 while 을 통해 반복시키고

c=fin.get() : 파일에서 문자를 읽게 만들어준다

while((c = fin.get()) != EOF) : 파일의 끝을 만나면 루프 종료

..... // 파일에서 읽은 값 c를 처리하는 코드

위의 처리를 통해 루프 종료한다

### 예제 12-5

char buf[81]; // 한 라인이 최대 80개의 문자로 구성된다고 가정

while(fin.getline(buf, 81)) { // 한 라인이 최대 80개의 문자로 구성

getline()을 이용하여 텍스트 파일을 읽는다

C++의 새로운 입출력 방식

get()/put()

- 문자 혹은 바이트 단위로 파일 입출력
- read()/write()
- 블록 단위로 파일 입출력

### 예제 12-10

fout.write((char\*)n, sizeof(n)); 과 fin.read((char\*)n, sizeof(n));

fout 과 fin 으로 배열을 쓰고 읽고

cout 으로 출력후

fin.close로 코드를 마친다

C++는 열려진 파일마다 두 개의 파일 포인터 유지

- get pointer : 파일 내에 다음에 읽을 위치
- put pointer : 파일 내에 다음에 쓸 위치

C++ 파일 입출력 방식

### **순차 접근**

- 읽은 다음 위치에서 읽고, 쓴 다음 위치에 쓰는 방식
- 디폴트 파일 입출력 방식

### **임의 접근**

- 파일 내의 임의의 위치로 옮겨 다니면서 읽고 쓸 수 있는 방식
- 파일 포인터를 옮겨 파일 입출력