

정리노트 #8

13조 (전대원, 소재현)

스트림 : 데이터의 흐름, 혹은 데이터를 전송하는 소프트웨어 모듈

C++ 스트림 종류

입력 스트림 : 입력 장치, 네트워크, 파일로부터 데이터를 프로그램으로 전달하는 스트림

출력 스트림 : 프로그램에서 출력되는 데이터를 출력 장치, 네트워크, 파일로 전달하는 스트림

출력 클래스들 정리

ios : 모든 입출력 스트림의 기본 클래스

istream, ostream, iostream : istream은 문자 단위 입력 스트림,

ostream은 문자 단위 출력 스트림, iostream은 문자 단위로 입출력을 동시에 할 수 있는 스트림 클래스

ifstream, ofstream, fstream : 파일에서 읽고 쓰는 기능을 가진 파일 입출력 스트림 클래스, ifstream은 읽을 때, ofstream은 쓸 때, fstream은 읽고 쓰기 동시에 할 때

입출력 스트림 객체 정리

- cin : istream 타입의 스트림 객체로서 키보드 장치와 연결
- cout : ostream 타입의 스트림 객체로서 스크린 장치와 연결
- cerr : ostream 타입의 스트림 객체로서 스크린 장치와 연결

오류 메시지를 출력할 목적, 스트림 내부 버퍼 거치지 않고 출력

- clog : Ostream 타입의 스트림 객체로서 스크린 장치와 연결, 오류 메시지를 출력할 목적, 스트림 내부에 버퍼 거쳐 출력

예제 11-1

```
#include <iostream>

using namespace std;

int main() {

// "Hi!"를 출력하고 다음 줄로 넘어간다.

cout.put('H');

cout.put('i');

cout.put(33);

cout.put('\n');

// "C++ "을 출력한다.

cout.put('C').put('+').put('+').put(' ');

char str[]="I love programming";

cout.write(str, 6); // str 배열의 6 개의 문자 "I love"를 스트림에 출력

}
```

- 아스키코드의 등장으로 33을 ! 문자로 사용
- Put() 메소드의 등장으로 연결하여 사용함

```
int ch; while((ch = cin.get()) != EOF) { // EOF 는 -1 cout.put(ch); // 읽은 문자 출력 if(ch == '\n')
break; // 키가 입력되면 읽기 중단 }
```

int get() 또한 나와서 문자들을 읽는다

예제 11-2

while((ch = cin.get()) != EOF) 에서 문자를 읽어준다 EOF 는 -1

get()이 읽는 도중 <Enter>키('\n')을 만날 때 주의

- 읽기를 중단하고 리턴
- <Enter> 키('\n')는 스트림 버퍼에 남아 있음
- 다시 get()으로 문자열 읽기를 시도하면 입력 스트림에 남은 '\n'키를 만나게 되어바로 리턴. 프로그램은 무한 루프에 빠질 수 있음
- 이때 이 문제를 해결하기 위해서는 cin.get()이나 cin.ignore(1);를 통해 문자 1개('\n')를 스트림에서 읽어 버려야 함

예제 11-3

cin.ignore(1); // 버퍼에 남아 있는 키 ('\n') 제거

위 부분을 통해 무한 루프에 빠지지 않게 함

입력 문자 건너 띄기와 문자 개수 알아내기

```
cin.ignore(10); // 입력 스트림에 입력된 문자 중 10개 제거
```

```
cin.ignore(10, ';'); // 입력 스트림에서 10개의 문자 제거. 제거 도중 ';'을 만나면 종료
```

읽은 문자 개수 리턴

```
char line[80];
```

```
cin.getline(line, 80);
```

```
int n = cin.gcount(); // 최근의 실행한 getline() 함수에서 읽은 문자의 개수 리턴
```

삽입 연산자(<<) 입출력과 같은 기호사용

- insertion operator, 삽입자라고도 부름
- << 연산자는 C++의 기본 연산자 : 정수 시프트 연산자
- ostream 클래스에 중복 작성되어 있음

```
class Point {
```

```
int x, y;
```

```
public:
```

```
Point(int x=0, int y=0) { this->x = x; this->y = y; }
```

```
};
```

```
Point p;
```

```
cin >> p;
```

```
cout << p;      cin >> p 실행하여 x좌표 >> 100, y좌표 >> 200
```

예제11-12

```
#include using namespace std;
```

```
ostream& fivestar(ostream& outs) { return outs << "*****";
```

```
} ostream& rightarrow(ostream& outs) { return outs << "---->";
```

```
} ostream& beep(ostream& outs) { return outs << '\a'; // 시스템 beep(뽕 소리) 발생 } int main()  
{ cout << "벨이 울립니다" << beep << endl; cout << "C" << rightarrow << "C++" << rightarrow  
<< "Java" << endl; cout << "Visual" << fivestar << "C++" << endl; }
```

stream클래스와 outs 를 활용하여 간단한 코드 작성