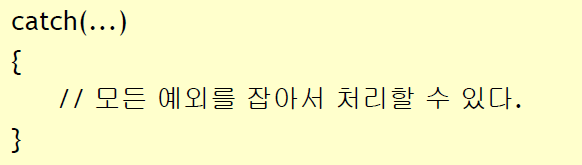
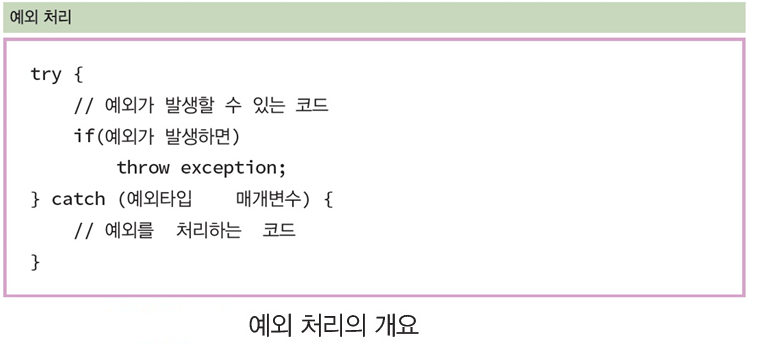
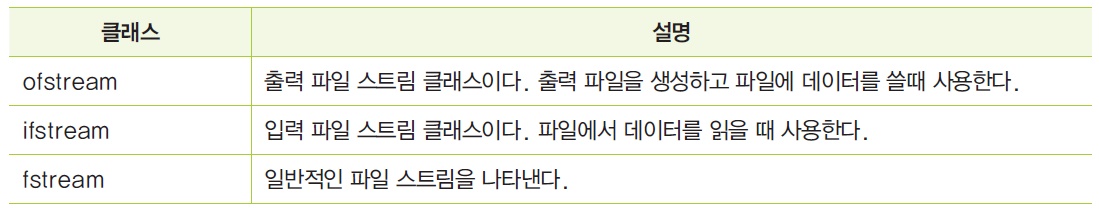
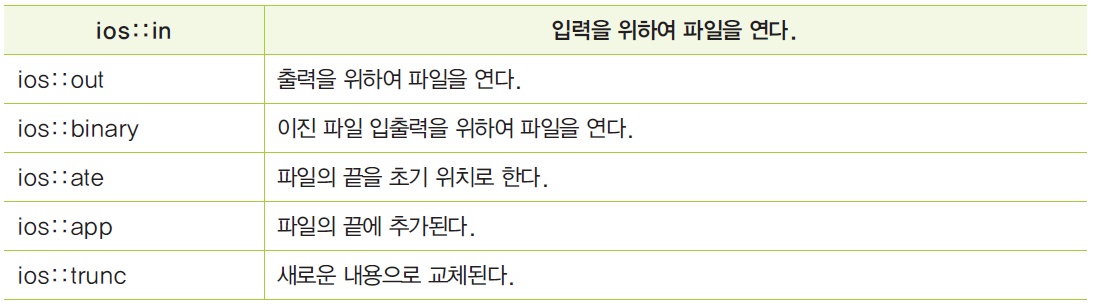
예외 (exception) : 잘못된 코드 , 부정확한 데이터 , 예외적인 상황에 의하여 발생하는 오류

인터럽트 : 컴퓨터 작동 중 문제가 발생한 경우라도 업무 처리가 계속될 수 있도록 하는 기능

예외는 함수를 넘어서 전달될 수 있다.

스트림(stream)은 “순서가 있는 데이터의 연속적인 흐름”

스트림은 입출력을 물의 흐름처럼 간주

#include <fstream>

쓰레드(thread)는 <thread>라이브러리에 정의

쓰레드(thread)란 프로세스내에서 실행되는 실행 단위

주로 프로그램에는 하나의 스레드로 구성되는데 프로그램은 여러 개의 스레드가 있음, (멀티스레드 )

멀티스레드를 실행하면 각 스레드들은 일정시간동안 번갈아가면서 실행

join()을 실행시키지 않으면 스레드가 종료되기 전에 프로그램이 끝나 런타임에러가 발생하므로 반드시 종료전에 join을 실행

void func1(int val) { //0, 1, 2, 가 번갈아 실행

for (int i = 0; i < 10000; i++) {

cout << val;

스레드t1과 t2의 join이 메인함수의 for문이전에 실행되었기때문에 스레드 t1과 t2가 종료되기를 기다렸다가 메인의 for문이 실행

t1이 먼저 종료되기때문에 0이 출력되기 전에 2가 반복적으로 나온다.

}

}

int main() {

thread t1(&func1, 1);

thread t2(&func1, 2);

for (int i = 0; i < 1000; i++) {

cout << 0;

}

t1.join();

t2.join();

return 0;

}

STL의 3가지 컴포넌트

컨테이너(container)

자료를 저장하는 구조이다.

벡터, 리스트, 맵, 집합, 큐, 스택과 같은 다양한 자료 구조들이 제공된다.

반복자(iterator)

컨테이너 안에 저장된 요소들을 순차적으로 처리하기 위한 컴포넌트

알고리즘(algorithm)

정렬이나 탐색과 같은 다양한 알고리즘을 구현  
  
순차 컨테이너:

자료를 순차적으로 저장

벡터(vector): 동적 배열처럼 동작한다. 뒤에서 자료들이 추가된다.

데크(deque): 벡터와 유사하지만 앞에서도 자료들이 추가될 수 있다.

리스트(list): 벡터와 유사하지만 중간에서 자료를 추가하는 연산이 효율적이다.  
  
연관 컨테이너

사전과 같은 구조를 사용하여서 자료를 저장

원소들을 검색하기 위한 키(key)

자료들은 정렬

집합(set): 중복이 없는 자료들이 정렬되어서 저장된다.

맵(map): 키-값(key-value)의 형식으로 저장된다. 키가 제시되면 해당되는 값을 찾을 수 있다.

다중-집합(multiset): 집합과 유사하지만 자료의 중복이 허용된다.

다중-맵(multimap): 맵과 유사하지만 키가 중복될 수 있다.

컨테이너 어댑터

순차 컨테이너에 제약을 가해서 데이터들이 정해진 방식으로만 입출력

스택(stack): 먼저 입력된 데이터가 나중에 출력되는 자료 구조

큐(queue): 데이터가 입력된 순서대로 출력되는 자료 구조

우선 순위큐(priority queue): 큐의 일종으로 큐의 요소들이 우선 순위를 가지고 있고 우선 순위가 높은 요소가 먼저 출력되는 자료 구조



상속 : 두 클래스 사이에 부모-자식의 상속관계를 선언한다. 부모를 기본클래스, 자식을 파생클래스

목적 : 간결한 클래스 작성, 클래스 간의 계층적 분류 및 관리에 용이함, 클래스 재사용과 확장을 통한 소프트웨어의 생산성 향상. 378