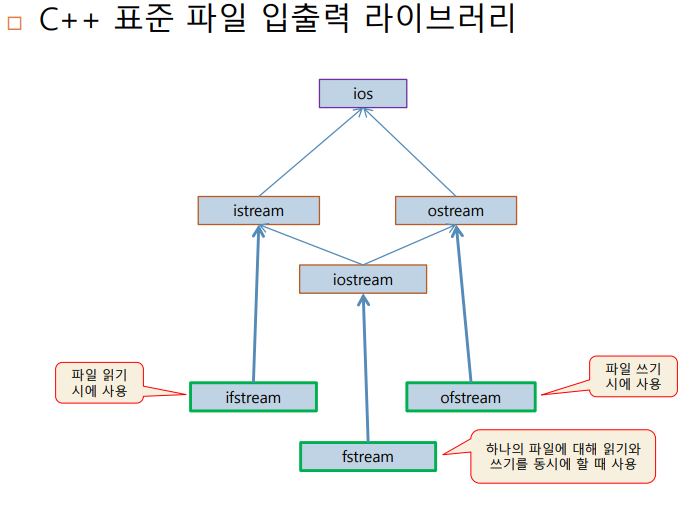
객체지향프로그래밍 (202004100 장주훈, 202084023 이병길)

**14주차 – C++ 파일 입출력, 예외 처리와 C언어와의 링크 지정**

텍스트 파일: 문자 만으로 구성된 문서

바이너리 파일: 문자, 그림, 표, 사운드, 동영산 등으로 구성된 문서



#include <iostream>

#include <fstream>은 따로 작성해야 함!

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| //예제 12-1  #include <iostream>  #include <fstream>  using namespace std;  int main() {  char name[10], dept[20];  int sid;  // 키보드로부터 읽기  cout << "이름>>";  cin >> name; // 키보드에서 이름 입력 받음  cout << "학번>>";  cin >> sid; // 키보드에서 학번 입력 받음  cout << "학과>>";  cin >> dept; // 키보드에서 학과 입력 받음  // 파일 열기. student.txt 파일을 열고, 출력 스트림 생성  ofstream fout("C:\\Temp\\student.txt");  if (!fout) { // 열기 실패 검사  cout << "C:\Temp\\student.txt 파일을 열 수 없다";  return 0;  }  // 파일 쓰기  fout << name << endl; // name 쓰기  fout << sid << endl; // sid 쓰기  fout << dept << endl; // dept 쓰기  fout.close(); // 파일 닫기  } |

텍스트, 폰트, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| #include <iostream>  #include <fstream>  using namespace std;  int main() {  // 스트림 객체 생성 및 파일 열기  ifstream fin;  fin.open("c:\\temp\\student.txt");  if (!fin) { // 파일 열기 실패 확인  cout << "파일을 열 수 없다";  return 0;  }  char name[10], dept[20];  int sid;  // 파일 읽기  fin >> name; // 파일에 있는 문자열을 읽어서 name 배열에 저장  fin >> sid; // 정수를 읽어서 sid 정수형 변수에 저장  fin >> dept; // 문자열을 읽고 dept 배열에 저장  // 읽은 텍스트를 화면에 출력  cout << name << endl;  cout << sid << endl;  cout << dept << endl;  fin.close(); // 파일 읽기를 마치고 파일을 닫는다.  } |

파일 모드: 파일 입출력에 대한 구체적인 작업 행태에 대한 지정

파일 모드 지정 – 파일 열 때

◼ open(“파일이름”, 파일모드) ◼ ifstream(“파일이름”, 파일모드), ◼ ofstream(“파일이름”, 파일모드)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

예제 12-3

|  |
| --- |
| #include <iostream>  #include <fstream>  using namespace std;  int main() {  const char\* file = "c:\\windows\\system.ini";  ifstream fin(file);  if (!fin) {  cout << file << " 열기 오류" << endl;  return 0;  }  int count = 0;  int c;  while ((c = fin.get()) != EOF) { // EOF를 만날 때까지 문자 읽기  cout << (char)c;  count++;  }  cout << "읽은 바이트 수는 " << count << endl;  fin.close(); // 파일 닫기  } |

텍스트 파일 연결하는 방법 예제 12-4

|  |
| --- |
| #include <iostream>  #include <fstream>  using namespace std;  int main() {  const char\* firstFile = "c:\\temp\\student.txt";  const char\* secondFile = "c:\\windows\\system.ini";  fstream fout(firstFile, ios::out | ios::app); // 쓰기 모드로 파일 열기  if (!fout) { // 열기 실패 검사  cout << firstFile << " 열기 오류";  return 0;  }  fstream fin(secondFile, ios::in); // 읽기 모드로 파일 열기  if (!fin) { // 열기 실패 검사  cout << secondFile << " 열기 오류";  return 0;  }  int c;  while ((c = fin.get()) != EOF) { // 파일 끝까지 문자 읽기  fout.put(c); // 읽은 문자 기록  }  fin.close(); // 입력 파일 닫기  fout.close(); // 출력 파일 닫기  } |

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

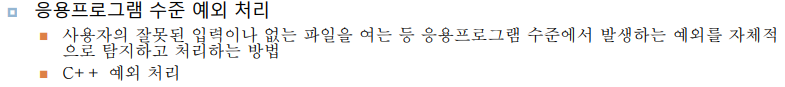
텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| //예제 12-8  #include <iostream>  #include <fstream>  using namespace std;  int main() {  const char\* file = "c:\\windows\\system.ini";  ifstream fin;  fin.open(file, ios::in | ios::binary); // 읽기 모드로 파일 열기  if (!fin) { // 열기 실패 검사  cout << "파일 열기 오류";  return 0;  }  int count = 0;  char s[32];  while (!fin.eof()) { // 파일 끝까지 읽는다.  fin.read(s, 32); // 최대 32 바이트를 읽어 배열 s에 저장  int n = fin.gcount(); // 실제 읽은 바이트 수 알아냄  cout.write(s, n); // 버퍼에 있는 n 개의 바이트를 화면에 출력  count += n;  }  cout << "읽은 바이트 수는 " << count << endl;  fin.close(); // 입력 파일 닫기  } |

12-9까지

13장 예외 처리



텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| //예제13-4  #include <iostream>  using namespace std;  int main() {  int n, sum, average;  while (true) {  cout << "합을 입력하세요>>";  cin >> sum;  cout << "인원수를 입력하세요>>";  cin >> n;  try {  if (n <= 0) // 오류 탐지  throw n; // 예외 발생. catch(int x) 블록으로 점프  else  average = sum / n;  }  catch (int x) {  cout << "예외 발생!! " << x << "으로 나눌 수 없음" << endl;  average = 0;  cout << endl;  continue;  }  cout << "평균 = " << average << endl << endl; // 평균 출력  }  } |

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명