컴퓨터와 프로그래밍

소프트웨어의 중요성 up

프로그래밍과 프로그래밍 언어 프로그래밍 컴퓨터가 처리할 일련의 작업묘사

프로그래밍언어 프로그래밍에 사용하는 컴퓨터 언어

기계어 0,1만 사용하는 가장 원시적인 언어

어셈블리어 기계어의 각 명령어를 기호와 일대일 대응시킨 언어

어셈블러 어셈블리어를 기계어로 변환

고급언어 사람이 이해하고 표현하기 쉬움

컴파일러 고급언어를 기계어로 변환

컴파일 고급언어를 기계어로 변환하는 과정

C++의 역사

흐름 BCPL 🡪 B언어 🡪 C언어 🡪 C++ 🡪 JAVA

표준 C++ 라이브러리 표준을 이용하면 모든 C++컴파일러로 컴파일 가능

C++의 특징

C++의 설계 목표 1. C언어로 작성된 프로그램과의 호환성

2. 객체 지향 개념 도입

3. 엄격한 타임체크를 통해 런타임에러를 줄이고 디버깅을 도움

4. 실행시간의 효율성 최소화

C언어에 추가한 기능 함수중복 동일한 함수 중복으로 선언가능(매개변수의 개수, 타입은 다름)

디폴트 매개 변수 매개변수가 기본값을 지니게 해줌

참조 변수에 포인터를 붙여 변수공간을 같이 사용

new,delete 연산자 동적메모리 할당, 해제

연산자 재정의 기존 연산자에 새로운 연산 정의

제네릭 함수∙클래스 함수, 클래스를 데이터 타입에 의존하지 않고 일반화

C++의 객체 지향 특성 캡슐화 데이터를 캡슐로 싸서 외부의 접근으로부터 데이터를 보호하는 객체 지향 특성

객체 클래스라는 틀에서 생겨난 인스턴스

클래스 class 키워드로 작성한 객체를 정의하는 틀

상속성 부모클래스가 자식클래스의 코드 상속 🡪 코드의 재사용성 증가

다형성 하나의 기능이 경우에 따라 서로 다르게 보이거나 다르게 작동 = 연산자 중복, 함수중복

객체도입의 목적 웨어의 생산성 향상, 쉬운 모델링

절차지향프로그래밍, 객체지향프로그래밍 절차지향프로그래밍 실행하고자 하는 절차대로 일련의 명령어를 나열

객체지향프로그래밍 실제 세상과 가깝게 모델링

제네릭 프로그래밍 제네릭 함수∙클래스 동일한 프로그래밍 코드에 다양한 데이터 타입을 적용할 수 있도록 함수, 클래스를 일반화 시킨 함수∙클래스

C++ 개발 과정

C++ 소스코드 작성

컴파일 C++ 소스코드가 문법에 맞게 작성되었는지 검사

기계어 코드로 변환해 목적파일 생성

링킹 목적파일이 참조하는 표준 라이브러리, 다른 객체의 함수, 객체, 데이터를 포함해 하나의 실행파일로 만드는 과정

결과로 .EXE파일 생성

프로그램실행과 디버깅 디버깅 프로그램 내에 오류 위치를 발견하거나 문제의 원인을 찾아 수정하는 과정

C++ 표준 라이브러리

개발자들이 불러 쓸 수 있는 다양한 종류의 함수와 클래스가 컴파일된 목적 파일

C라이브러리 기존 C 표준 라이브러리

C로 시작하는 헤더파일

C++입출력 라이브러리 콘솔 및 파일 입출력을 위한 함수와 클래스

제네릭 프로그래밍을 지원하기 위해 템플릿으로 작성

C++ STL 라이브러리 제네릭 프로그래밍을 지원하기 위한 함수, 클래스

비주얼 스튜디오를 이용한 C++ 개발

C++ 통합 개발 환경

비주얼 스튜디오 시작

프로젝트 만들기

C++소스코드 편집

컴파일과 빌드

프로그램 실행