1. C++ 시작

프로그래밍 언어

- 기계어
  - 0,1 이진수로 구성
  - CPU 는 기계어만 처리가능
- 어셈블리어
- 고급언어
  - C/C++ 같은거

컴파일러 : 고급 언어로 작성된 프로그램 → 기계어

C++ —(컴파일러)→ 어셈블리어 —(어셈블러)→ 기계어

C++ 설계 목적

- C 언어와 호환성
  - C 언어의 문법 체계를 계승
    - C 프로그램 그대로 사용 O
    - 라이브러리 사용 가능
- 객체지향
  - 캡슐화, 상속, 다형성
  - 소프트웨어 재사용 → 생산성 향상
  - 작성, 관리, 유지보수 쉬움
- 엄격한 타입 체크
- 실행 시간의 효율성 저하 최소화
  - 인라인 함수.

- 캡슐화
  - 외부 접근으로부터 보호
  - class 로 표현
- 클래스와 객체
  - 클래스 : 객체 만드는 틀
  - 객체 (실체) : 틀에서 생겨난 실체
- 상속성
- 다형성

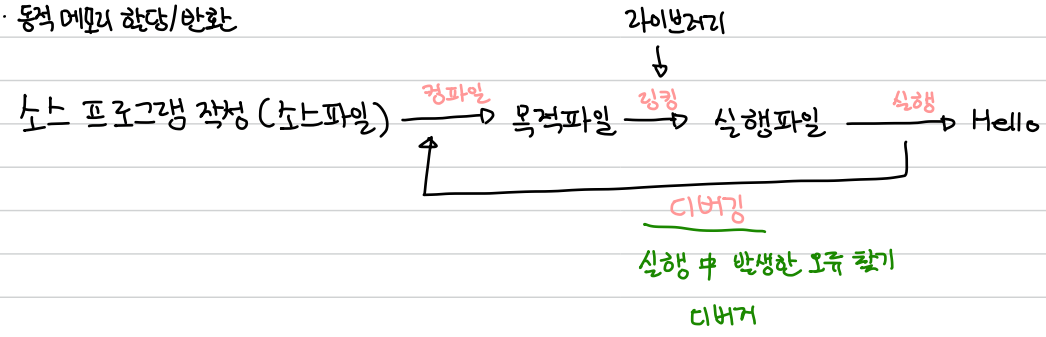
  하나의 기능이 경우에 따라 다르게 작동

  → 연산자 중복, 함수중복, 함수 재정의

C 언어에 추가한 기능

- 함수 중복
  - 동일한 이름의 함수 선언 가능
- 디폴트 매개변수
  - 매개 변수에 디폴트 값 전달
- 참조 · 참조변수
- 참조에 의한 호출
- new/delete 연산자
  - 동적 메모리 할당/반환

C++ 프로그램 개발과정

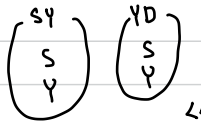


2. C++ 프로그래밍의 기본

- main() 표준모양

```
int main() {
    return 0; // 생략가능
}
```

- #include <iostream>
  - 전처리기에 내리는 지시
    - 컴파일전 헤더파일 소스에 확장
- <iostream> 헤더파일
  - 표준 입출력 위한 클래스, 객체, 변수 선언
    - cout, cin, <<, >> 등
      - 연산자.
- namespace
  - 이름 공간



서로 다른거

<사용법>

이름공간:: 이름

- std::

C++ 표준에서 정의한 namespace 中 하나.

using namespace std; → std:: 생략

- Cin

입력 버퍼 내장 <Enter>까지

- 변수 선언

아무때나 가능

장: 변수 이름 타이핑 오류 ↓

단: 변수 한 눈에 보기 어려움, 찾기 어려움

- 문자열

C-스트링 문자열

→ <string> <cstring>이 표준

단는 문자 배열

- 공백이 낀 문자열

cin.getline(char buf[], int size, char delimit Char)

buf에 최대 size-1개 문자입력, delimit → 디폴트='\n'

char 배서 종단

- String

C++ 표준 클래스

복사·비교·수정 함수 제공

객체지향적

getline(cin, 이름, char delimit)

도가능

- 헤더파일

include <헤더파일>

컴파일러가 설치된 폴더에서 찾기

include " 헤더파일"

프로젝트 폴더 or include 폴더에서 찾기

헤더파일 - 함수의 원형이 들어있다.

3. 클래스와 객체

- 객체의 캡슐화

· 목적

객체 내 데이터에 대한 보안보호, 비 직접적인 일부 요소만 공개

멤버 함수 / 멤버 변수
(행동) (상태)

객체이름.변수=0 객체이름.함수

- 클래스와 객체

· 클래스
- 객체의 설계도
- 객체 x 실체 x
- 멤버 변수+멤버함수 선언

· 객체
멤버 변수+멤버 함수 구성

- 클래스

<선언부>
Class 이용해서 클래스 선언
멤버 변수·함수 선언
멤버변수 초기화 x
멤버 함수 원형형태 선언
멤버 접근 권한 선정
public protected private
(디폴트)

<구현부>
구현하기

```
class circle {
public:
    int radius;
    double getArea();
};
double Circle::getArea() {
    return 3.14 * radius * radius;
}
```

- 생성자

객체가 생성되는 시점에서 자동으로 호출되는 멤버함수
클래스 이름과 동일한 멤버 함수
생성자 코드에서 멤버 변수 초기화 가능
생성자 x → 기본 생성자 자동 생성

- 소멸자

리턴 타입 x
클래스 이름 앞에 ~
하나만
자동 생성
없으면

★ ★ ★
객체는 생성의 반대 순으로 소멸
동적 → new/delete

- 접근 지정자

class private → 동일한 클래스의 멤버 함수에만 허용
구조체 public → 모든 다른 클래스에 허용
protected → 자신+상속받은 클래스

- 분리 컴파일

클래스 재사용
헤더파일에
#ifndef 헤더파일이름대문자_H
#define 헤더파일이름대문자_H
⋮
#endif

- 인라인 함수

프로그램 실행속도 개선
x 긴 함수, static 변수, 반복문, switch문, goto문 x
전체 코드 크기 증가
짧은 코드 함수

4. 객체 포인터와 객체 배열, 객체의 동적 생성

- 객체 포인터

객체 *p;

p = &객체 이름;

p-> getArea();

(*p). getArea();

- 객체 배열

Circle c[3];

매개변수 없는 생성자 호출

초기화

Circle circleArray[3] = {Circle(10), Circle(20), Circle()};

- 동적 메모리 할당 및 반환

<new / delete>

데이터 타입 *포인터 변수 = new 데이터 타입;

delete 포인터 변수

동적할당이 아닌 메모리반환 } 오류
두번 반환 }

- 배열 동적할당 및 반환

데이터타입 * 포인터변수 = new 데이터 타입 [배열의 크기];

delete [] 포인터 변수

- 객체 동적할당 및 반환

클래스이름 * 포인터변수 = new 클래스이름;

클래스이름 * 포인터 변수 = new 클래스이름(생성자 매개변수 리스트);

delete 포인터 변수;

- 객체 배열 동적 생성 및 반환

클래스이름 * 포인터변수 = new 클래스이름 [배열 크기];

delete [] 포인터변수 ;

- this 포인터

클래스 멤버 함수 내에서만 사용

컴파일러가 선언한 변수

- 매개변수 이름과 멤버 변수의 이름이 같은경우
- 멤버 함수가 객체 자신의 주소를 리턴할때

* 멤버함수 x -> 사용 못함

static 멤버함수에서 사용 불가