객체지향프로그래밍 (202004100 장주훈, 202084023 이병길)

**11주차** – 가상함수와 추상클래스

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명



오버라이딩 사용 이유: 가독성, 함수의 재사용 등 효율이 올라감.



텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 실행결과 Base::f() called  Derived::f() called | Derived::f() called Derived::f() called |

함수 재정의는 각각 컴파일 타입에 지정을 해서 함수를 두 개 존재하게 만듬.

오버라이딩은 런타임시 바인딩을 함.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

오버라이딩의 목적 – 파생 클래스에서 구현할 함수 인터페이스 제공(파생 클래스의 다형성)

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

동적바인딩의 목적: 파생클래스

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

오버라이딩의 성공 조건: 가상 함수 이름, 매개 변수 타입과 개수, 리턴 타입이 모두 일치

오버라이딩 시 virtual 지시어 생략 가능 - 가상 함수의 virtual 지시어는 상속됨, 파생 클래스에서 virtual 생략 가능

가상 함수의 접근 지정 - private, protected, public 중 자유롭게 지정 가능

기본 클래스를 호출하고 싶을 때, **기본클래스::가상함수()** 형태로 기본 클래스의 가상 함수를 정적 바인딩으로 호출

가상 소멸자: 소멸자를 virtual 키워드로 선언(소멸자 호출 시 동적 바인딩 발생)

텍스트, 폰트, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
|  | 실행 결과를 예측하여 쓰시오.  ~Derived()  ~Base()  ~Derived()  ~Base() |

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명



과제 – 오버라이딩을 이용한 연습문제

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

파생 클래스에서 가상 함수 draw()의 재정의

* 동적 바인딩에 의해 어떤 경우에도 자신이 만든 draw()가 호출됨을 보장 받음

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

동적바인딩 실행 부분 자세히 보기!

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=lghlove0509&logNo=220955178807> c언어와 비교해보기!

순수 가상 함수: 함수의 코드가 없고 선언만 있는 가상 멤버 함수

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명



목적: 상속에서 기본 클래스의 역할을 하기 위함(순수 가상 함수를 통해 파생 클래스에서 구현할 함수의 형태(원형)을 보여주는 인터페이스 역할)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

예제 9-7

|  |  |
| --- | --- |
| Class Adder : public Calculator{  protected:  int calc(int a, int b){  return a+b;  }  } | Class Subtractor : public Calculator{  protected:  int calc(int a, int b){  return a-b;  }  } |