virtual
abstract
interface

남동강

가상 함수(virtual function)

- **virtual 키워드**를 이용하여 선언한다
- **자식 클래스**에서 **가상 함수를 재정의** 할 수 있고, 재정의된 **자식의 멤버 함수 또한 가상 함수**로 분류 된다

```
class Person
{
    …
    virtual void show() { std::cout << "person" << std::endl; }
};

class Student : public Person
{
    …
    virtual void show() { std::cout << "student" << std::endl; }
};
```

순수 가상 함수(pure virtual function)

- 가상 함수는 기본적으로 **선언 후 정의를 하지 않아도** 되며, 이런 가상 함수를 **순수 가상 함수**라 한다
- **순수 가상 함수**는 **자식 클래스**에서 **반드시 재정의** 하여야 한다

```
class Person
{
    …
    virtual void show() = 0;       //or
    virtual void show() = NULL;    //or
    virtual void show() abstract;

};
class Student : public Person
{
    …
    virtual void show() { std::cout << "student" << std::endl; }
};
```

추상 클래스(abstract class)

- **하나 이상의 순수 가상 함수를 포함하는 클래스**
- 해당 클래스는 **인스턴스를 만들 수 없다**
  - **인스턴스**:클래스를 이용해 만들어낸 **객체**
- 추상 클래스의 인스턴스는 자식을 통하여 만들 수 있다

```
class Person
{
    …
    virtual void show() abstract;
};
class Student : public Person
{
    …
    virtual void show() { … }
};
Person* p = new Student;
p->show();
delete p;
```

추상 클래스(abstract class)

- **추상 클래스**는 기본 class 형에 순수 가상 함수를 포함하는 것으로 따로 **기본 클래스와 구분** 짓지 않았으나 **abstract 키워드**가 생기며 이 키워드를 붙이는 것으로 구분 짓게 되었다

```
class Person abstract
{
    virtual void show() abstract;
};
```

- Interface

- 순수 가상 함수만 선언된 클래스
- 생성자, 소멸자, 연산자를 포함할 수 없다
- 클래스와 달리 기본 속성을 public으로 가진다
- abstract와 달리 표준 키워드가 아니다

```
__interface INTERFACE
{
    void func1();
};
class A : public INTERFACE
{
    virtual void func1() { … }
};
```

virtual 소멸자

- 일반적인 상속관계의 생성자와 소멸자의 흐름

부모생성자->자식생성자->자식소멸자->부모소멸자

- 순수 가상 함수를 포함하는 상속 관계에서의 흐름

부모생성자->자식생성자->부모소멸자

- 부모의 소멸자에 virtual 키워드를 추가하여 문제를 해결 할 수 있다

```
class Person
{
    ...
    virtual~Person() { std::cout << "~person" << std::endl; }
};
class Student : public Person
{
    ...
    virtual ~Student() { std::cout << "~student" << std::endl; }
};
```

interface

- __interface는 아직 표준화 되지 않은 키워드
- 일반적으로 추상 클래스가 상속을 받아 사용한다
- Interface는 기본 클래스에서 상속 받을 수 없다
- 공용 순수 가상 함수만을 포함할 수 있다
- 생성자, 소멸자를 포함할 수 없다
- 정적(static) 함수는 포함할 수 없다
- 멤버 변수를 포함할 수 없다