Core Concepts of OOP (Object-oriented Programming)

ESM2014-41 객체지향프로그래밍 및 실습 SKKU 시스템경영공학과 조영일

- 00P Example of Movie Ticketing System
- 우리는, 특정 영화 프랜차이즈의 영화 좌석 예매 시스템을 만들어야합니다. 고객은 본인이 시청할 영화의 상영관과 상영 시간과 좌석을 선택 하고 예약 할 수 있습니다.

- 00P Example of Movie Ticketing System
- Classes / Attribute / Method(Encapsulation)

Theater Class

name address tel_no movies

add_movie()
remove_movie()

Movie Class

name genre director rating status

update_rating()
change_status()

Customer Class

name
email
mobile_num
date_of_birth
grade

check_age()
get_benefit()

Booking Class

theater
movie
time
customer
payment_status

change_time()
change_status()

- 00P Example of Movie Ticketing System
- Objects

Theater Class

name address tel_no movies

add_movie()
remove_movie()

센트럴점 Object (Instance of Theater Class)

name : 센트럴점

address : 서울시 강남구 서초동

tel_no : 02-2222-2222

movies : [list of movie objects]

add_movie()
remove_movie()

성균관대점 Object (Instance of Theater Class)

name : 성균관대점

Address : 경기도 수원시 장안구

Tel_no : 031-2222-2222

movies : [list of movie objects]

add_movie()
remove_movie()

- 00P Example of Movie Ticketing System
- Objects

Customer Class

name email mobile_num date_of_birth grade

check_age()
get_benefit()

나성균 Object (Instance of Customer Class)

name : 나성균

email : me@skku.edu

mobile_num : 010-1111-2222 date of birth : 20010920

grade : vip

check_age()
get_benefit()

너관대 Object (Instance of Theater Class)

name : 너관대

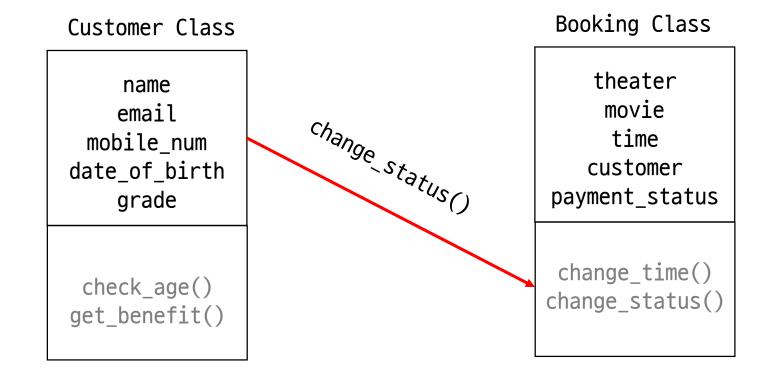
email : you@skku.edu

Mobile_num : 010-2222-2222 date of birth : 20031020

grade : silver

check_age()
get_benefit()

- 00P Example of Movie Ticketing System
- Interaction Between Objects



- Encapsulation
- 객체 간의 상호 작용은 상대 객체에 공개되어 있는 Method를 통해서만 수행한다.
- Interface = Method = Message
- Encapsulation의 특징 : 정보 은닉(Data-Hiding), 객체 내부의 구체적 인 구현은 외부에 최대한 숨기고 필요한 부분만(Interface) 공개한다.
- -이를 통해 프로그램을 구성하는 요소들 간의 결합성을 최소화 시키고 (Decoupling) 각 요소들이 독립적으로 유지되게 함으로서 각 요소의 유지보수 편의성, 재사용성을 높일 수 있다.

• 00P Example of Movie Ticketing System

- Inheritance Customer Class (Parent Class) IS-A Relation name email Employee Customer is a Customer mobile num Bad Customer is a Customer. date_of_birth grade check age() get benefit() Employee Customer Class Bad Customer Class (Child Class) (Child Class) name name email email mobile num mobile num date_of_birth date_of_birth grade grade employee_num comments check age() check age() get benefit() get benefit() get employee num() add comments()

- 00P Example of Movie Ticketing System
- Polymorphism
- Same name, different function(e.g. add 10% more discount benefit to employee customer)

Customer Class
(Parent Class)

name
email
mobile_num
date_of_birth
grade

check_age()
get_benefit()

Employee Customer Class (Child Class)

name email mobile_num date_of_birth grade employee_num

check_age()
 get_benefit()
get_employee_num()

• 객체 지향 프로그래밍 방법론은 특정한 기술이나 코드 작성법이 아니다. 현실 세계의 문제들을 프로그램 속에 반영하기 위한 개념체계이다.