과제 02 보고서

모바일 앱 개발 기초반

10413 박희찬

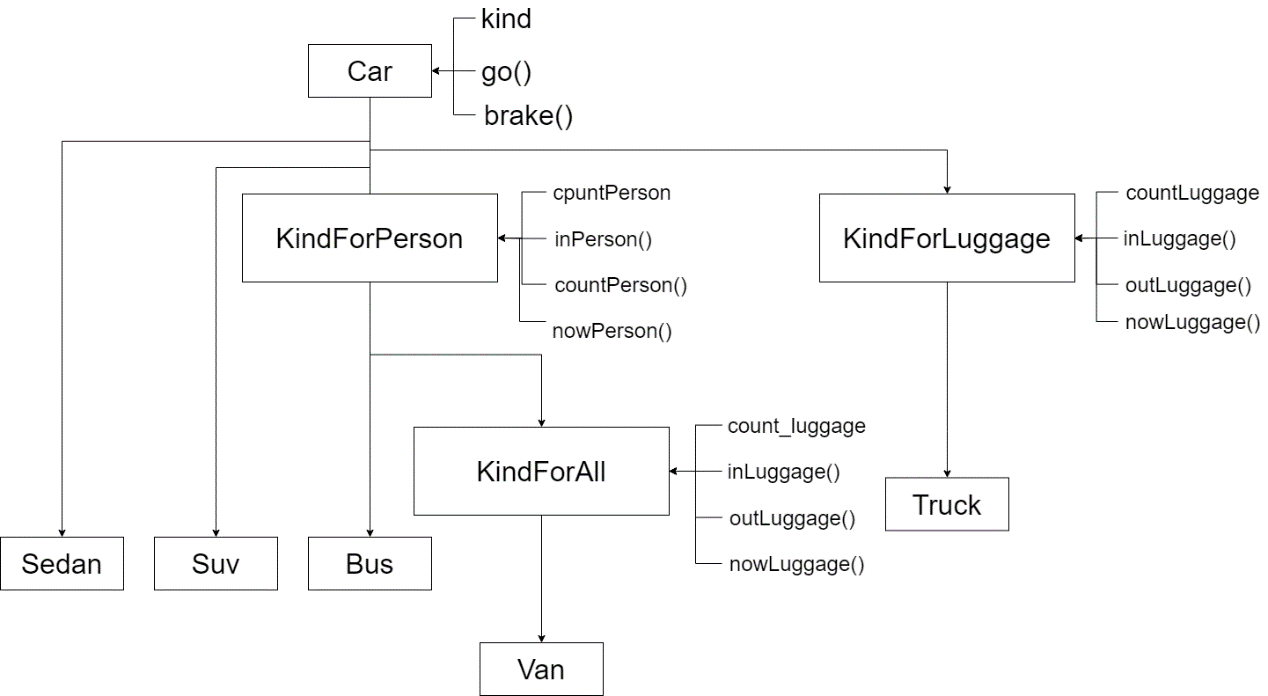
# **목차**

# **클래스 구조**

# **코드 설명**

# **코드의 문제점**

# 클래스 구조



<클래스 구조, 상속 관계>

## Car 클래스

최상위인 Car 클래스를 만들어 주었고 다른 자동차들은 모두 이 클래스를 상속받게 만들어주었습니다.

멤버 변수는 자동차의 종류를 저장하는 kind

맨버 함수는 주행, 브레이크 기능이 있는 go(), brake()함수가 있습니다.

추상 클래스로 만들어주어서 kind를 자동차의 종류 클래스에서 상속을 받을 때 값을 지정해줍니다.

## KindForPerson 클래스

사람을 태우는 자동차가 상속받는 클래스, Car 클래스를 상속받아서 kind와 go(), brake()를 가지고 있습니다.

멤버 변수는 현재 탑승자 수를 저장한 countPerson, 멤버 함수로는 사람을 태우는 inPerson(), 사람을 내리는 outPErson()을 가지고 있습니다.

## kindForLuggage 클래스

짐을 싣는 자동차가 상속받는 클래스, Car 클래스를 상속받아서 kind, go(), brake()를 가지고 있습니다.

멤버 변수로는 현재 짐의 수를 저장하는 countLuggage, 멤버 함수로는 짐을 싣는 inLuggage(), 짐을 내리는 outLuggage()를 가지고 있습니다.

## kindForAll 클래스

사람과 짐 모두 운반하는 자동차가 상속받습니다. kindForPerson 클래스를 상속받아서 kind, go(), brake() 를 가지고 있습니다.

멤버 변수와 멤버 함수로 짐을 싣고 내리는 기능인 countLuggage, inLuggage(), outLuggage()를 가지고 있고 이때 기능은 kindForLuggage클래스에서 정의해준 기능과 동일합니다.

## Sedan 클래스

자동차 종류 중 세단 자동차 클래스, Car 클래스를 상속받습니다.

## Suv 클래스

자동차 종류 중 SUV를 구현한 클래스, Car크래스를 상속받습니다.

## Bus 클래스

자동차 종류 중 버스를 구현한 클래스, 사람을 태우기 때문에 kindForPerson 클래스를 상속받습니다.

## Van 클래스

자동차 종류 중 승합차를 구현한 클래스 사람과 짐 모두를 운반하기 때문에 kindForAll 클래스를 상속받습니다.

## Truck 클래스

자동차 종류 중 트럭을 구현한 클래스, 짐을 운반하기 때문에 kindForLuggage 클래스를 상속받습니다.

# 코드 설명

자동차 종류 하나는 하나의 클래스(Sedan, Suv, Bus, Van, Truck 클래스)로 구현되었고 각자의 역할에 따라 상속받는 부모 클래스가 달라집니다.

모든 자동차 종류 클래스는 Car 클래스의 자손 클래스인데 Car클래스에서 자동차의 종류, go, brake 기능이 만들어졌습니다. Car클래스는ㄴ 추상화 클래스여서 kind를 선언해주기만 하고 할당해주지는 않습니다.

사람을 태우는 자동차는 KindForPerson 클래스를 상속받는데 이때 이 클래스는 Car클래스를 상속받아서 kind, go(), brake()를 가지고 있습니다. 추상화 클래스로 만들어주어서 kind에 값을 할당해주지 않아도 됩니다.

짐을 사람을 태우는 기능들을 만들어서 이 클래스를 상속받는 자동차는 사람을 태우고 내릴 수 있습니다.

짐을 싣는 자동차의 클래스는 KindForLuggage 클래스를 상속받는데 이 클래스도 Car클래스를 상속받아서 자동차 종류와 기본적인 주행, 브레이크 기능을 사용할 수 있고 이 클래스 역시 추상화 클래스여서 kind의 값을 할당하지 않아도 됩니다.

짐을 싣는 기능을 만들어 주어서 이 클래스를 상속받은 자동차들은 짐을 싣고 내릴 수 있습니다.

사람과 짐을 모두 싣는 자동차는 kindForAll 상속받는데 이 클래스는 KindForPerson 클래스를 상속받아서 자동차의 종류와 기본 기능, 사람을 태우는 기능을 사용할 수 있고

멤버 변수와 함수로 KindForLuggage와 동일한 기능을 하는 짐을 싣는 기능을 만들어 주어서 이 클래스를 상속받은 클래스는 사람을 태우는 기능과 짐을 싣는 기능을 모두 사용할 수 있습니다.

각 자동차의 종류 클래스는 각자의 종류에 맞게 사람을 태운다면 KindForPerson 클래스를,

짐을 싣는다면 KindForLuggage 클래스를, 짐과 사람 모두를 태운다면 KindForPErson 클래스를

기본적인 자동차라면 Car클래스를 상속받아서 만들어줍니다.

Driver클래스는 driving 함수를 가지고 있는데 인수로 Car의 객체를 받고 받은 객체의 go() 함수를 실행시켜줍니다.

모든 자동차 클래스는 Car의 자손 클래스여서 함수의 인수로 전달이 가능하고 go() 함수를 실행하시었습니다.

# 코드의 문제점

이 코드의 가장 큰 문제점은 KindForAll 클래스에서 KindForLuggage 클래스에서 만들어준 기능들을 똑같이 다시 만들어 준다는 것입니다.

이렇게 코드를 작성한 이유는 코틀린은 하나의 클래스에 2개 이상의 클래스를 상속하는 게 불가능 하기 때문이고

다중상속이 가능한 인터페이스를 사용하지 않고 추상화 클래스를 사용한 이유는 멤버 변수를 사용하기 위해 사용하였습니다. Car 클래스의 자동차의 종류를 저장하는 kind변수를 만들어주었는데 Car 클래스에서 만들어주어 모든 자동차 종류 클래스가 kind를 가질 수 있도록 만들어주었습니다.