모바일프로그래밍 과제04 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

예제3-12

[소스코드]

*/\* 예제3-12 Car.kt-정적 구성 요소 추가 \*/  
/\* 04 정적 필드, 정적 메소드, 상수 필드 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
 companion object {  
 var carCount : Int = 0  
 const val MAXSPEED : Int = 200  
 const val MINSPEED : Int = 0  
 fun currentCarCount() : Int {  
 return carCount  
 }  
 }  
  
 constructor(color: String, speed: Int) {  
 this.color = color  
 this.speed = speed  
 }  
  
 constructor(speed: Int) {  
 this.speed = speed  
 }  
  
 constructor() {  
 }  
  
 fun upSpeed(value: Int) {  
 if(speed+value >= 200)  
 speed = 200  
 else  
 speed = speed + value  
 }  
 fun downSpeed(value: Int) {  
 if(speed-value <= 0)  
 speed = 0  
 else  
 speed = speed - value  
 }  
}

예제3-13

[소스코드]

*/\* 예제3-13 exam07.kt \*/  
/\* 04 정적 필드, 정적 메소드, 상수 필드 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
 companion object {  
 var carCount : Int = 0  
 const val MAXSPEED : Int = 200  
 const val MINSPEED : Int = 0  
 fun currentCarCount() : Int {  
 return carCount  
 }  
 }  
  
 constructor(color: String, speed: Int) {  
 this.color = color  
 this.speed = speed  
 }  
  
 constructor(speed: Int) {  
 this.speed = speed  
 }  
  
 constructor() {  
 }  
  
 fun upSpeed(value: Int) {  
 if(speed+value >= 200)  
 speed = 200  
 else  
 speed = speed + value  
 }  
 fun downSpeed(value: Int) {  
 if(speed-value <= 0)  
 speed = 0  
 else  
 speed = speed - value  
 }  
}  
fun main() {  
 var myCar1 : Car = Car("빨강", 0)  
 var myCar2 : Car = Car("파랑", 0)  
 var myCar3 : Car = Car("초록", 0)  
  
 *println*("생상된 차의 대수(정적 필드) ==> " + Car.carCount)  
 *println*("생산된 차의 대수(정적 메소드) ==> " + Car.currentCarCount())  
 *println*("차의 최고 제한 속도 ==> " + Car.MAXSPEED)  
  
 *println*("PI의 값 ==> " +Math.*PI*)  
 *println*("3의 5제곱 ==> " + Math.pow(3.0, 5.0))  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-15

[소스코드]

*/\* 예제3-15 exam08.kt \*/  
/\* 01 클래스 상속과 메소드 오버라이딩 \*/*open class Car {  
 var color : String = ""  
 var speed : Int = 0  
 companion object {  
 var carCount : Int = 0  
 const val MAXSPEED : Int = 200  
 const val MINSPEED : Int = 0  
 fun currentCarCount() : Int {  
 return carCount  
 }  
 }  
  
 constructor(color: String, speed: Int) {  
 this.color = color  
 this.speed = speed  
 }  
  
 constructor(speed: Int) {  
 this.speed = speed  
 }  
  
 constructor() {  
 }  
  
 open fun upSpeed(value: Int) {  
 if(speed+value >= 200)  
 speed = 200  
 else  
 speed = speed + value  
 }  
 fun downSpeed(value: Int) {  
 if(speed-value <= 0)  
 speed = 0  
 else  
 speed = speed - value  
 }  
}  
  
class Automobile : Car {  
 var seatNum : Int = 0  
  
 constructor() {  
 }  
 fun countSeatNum() : Int {  
 return seatNum  
 }  
  
 override fun upSpeed(value: Int) {  
 if(speed+value >= 300)  
 speed = 300  
 else  
 speed = speed + value  
 }  
}  
  
fun main() {  
 var myCar1 : Car = Car("빨강", 0)  
 var auto : Automobile = Automobile()  
 auto.upSpeed(250)  
 myCar1.upSpeed(250)  
 *println*("승용차(Automobile)의 속도는 " + auto.speed + "km입니다.")  
 *println*("차(Car)의 속도는 " + myCar1.speed + "km입니다.")  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-18

[소스코드]

*/\* 예제3-18 exam10.kt \*/  
/\* 04 인터페이스와 다중 상속 \*/*abstract class Animal {  
 var name : String = ""  
 abstract fun move()  
}  
  
interface iAnimal {  
 abstract fun eat()  
}  
  
class iCat : iAnimal {  
 override fun eat() {  
 *println*("생선을 좋아한다.")  
 }  
}  
  
class iTiger : Animal(), iAnimal {  
 override fun move() {  
 *println*("네 발로 이동한다.")  
 }  
 override fun eat() {  
 *println*("멧돼지를 잡아먹는다.")  
 }  
}  
  
class Eagle : Animal() {  
 var home : String = ""  
 override fun move() {  
 *println*("날개로 날아간다.")  
 }  
}  
  
fun main() {  
 var cat : iCat = iCat()  
 cat.eat()  
  
 var tiger : iTiger = iTiger()  
 tiger.move()  
 tiger.eat()  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제4-1

[소스코드]

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 /\* 02 View 클래스의 XML 속성  
 예제4-1 id 속성의 XML 코드 \*/  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="성별 선택" />  
  
 <RadioButton  
 android:id="@+id/female"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="여성" />  
  
 <RadioButton  
 android:id="@+id/male"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="남성" />  
</LinearLayout>

[실행결과]



예제4-3

[소스코드]

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 /\* 02 View 클래스의 XML 속성  
 예제4-3 layout\_width, layout\_height 속성의 XML 코드2 \*/  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="버튼입니다" />  
</LinearLayout>

[실행결과]



예제4-6

[소스코드]

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 /\* 02 View 클래스의 XML 속성  
 예제4-6 layout\_width, layout\_height 속성의 XML 코드5 \*/  
 <Button  
 android:layout\_width="1080px"  
 android:layout\_height="1920px"  
 android:text="버튼입니다" />  
</LinearLayout>

[실행결과]

