모바일프로그래밍 과제03 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

예제3-2

[소스코드]

*/\* 예제3-2 exam01.kt \*/  
/\* 01 변수와 데이터 형식 \*/  
/\* 정수형 변수, 실수형 변수, 문자형 변수, 문자열 변수를 선언하고 값을 대입한 후 출력하는 예제 \*/*fun main() {  
 var var1 : Int = 10 */\* var var1 = 10 \*/* var var2 : Float = 10.1f */\* var var2 = 10.1f \*/* var var3 : Double = 10.2 */\* var var3 = 10.2 \*/* var var4 : Char = '안' */\* var var4 = '안' \*/* var var5 : String = "안드로이드" */\* var var5 = "안드로이드" \*/  
 println*(var1)  
 *println*(var2)  
 *println*(var3)  
 *println*(var4)  
 *println*(var5)  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-3

[소스코드]

*/\* 예제3-3 exam02.kt \*/  
/\* 02 조건문: if, when \*/*fun main() {  
 var count : Int = 85  
 if(count >= 90) {  
 *println*("if문: 합격 (장학생)")  
 } else if(count >= 60) {  
 *println*("if문: 합격")  
 } else {  
 *println*("불합격")  
 }  
  
 var jumsu : Int = (count / 10) \* 10  
 when(jumsu) {  
 100 -> *println*("when문: 합격(장학생)") */\* in 90 .. 100 -> println("when문: 합격(장학생)") \*/* 90 -> *println*("when문: 합격(장학생)")  
 80, 70, 60 -> *println*("when문: 합격") */\* in 60 .. 89 -> println("when문: 합격") \*/* else -> *println*("when문: 불합격")  
 }  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-4

[소스코드]

*/\* 예제3-4 exam03.kt \*/  
/\* 04 반복문: for, while \*/*fun main() {  
 var one : IntArray = *intArrayOf*(10,20,30)  
 for(i in one.*indices*) {  
 *println*(one[i])  
 }  
 for(value in one) {  
 *println*(value)  
 }  
  
 var two : Array<String> = *arrayOf*("하나", "둘", "셋")  
 for(i in 0..2 *step* 1) {  
 *println*(two[i])  
 }  
 var k : Int = 0  
 while(k < two.size) {  
 *println*(two[k])  
 k++  
 }  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-5

[소스코드]

*/\* 예제3-5 exam04.kt \*/  
/\* 05 메소드와 전역변수, 지역변수 \*/*var *myVar* : Int = 100  
fun main() {  
 var myVar : Int = 0  
 *println*(myVar)  
  
 var sum : Int = *addFunction*(10,20)  
 *println*(sum)  
}  
  
fun addFunction(num1: Int, num2: Int) : Int {  
 var hap : Int  
 hap = num1 + num2 + *myVar* return hap  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-6

[소스코드]

*/\* 예제3-6 exam05.kt \*/  
/\* 06 예외 처리: try~catch \*/*fun main() {  
 var num1 : Int = 100  
 var num2 : Int = 0  
 try {  
 *println*(num1/num2)  
 } catch (e : ArithmeticException) {  
 *println*("계산에 문제가 있습니다")  
 }  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-7

[소스코드]

*/\* 예제3-7 Car.kt-기본 구조 \*/  
/\* 01 클래스 정의와 인스턴스 생성 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
  
 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  
 }  
}

예제3-8

[소스코드]

*/\* 예제3-8 exam06.kt \*/  
/\* 01 클래스 정의와 인스턴스 생성 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
  
 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()  
 myCar1.color = "빨강"  
 myCar1.speed = 0  
  
 var myCar2 : Car = Car()  
 myCar2.color = "파랑"  
 myCar2.speed = 0  
  
 var myCar3 : Car = Car()  
 myCar3.color = "초록"  
 myCar3.speed = 0  
  
 myCar1.upSpeed(50)  
 *println*("자동차1의 색상은 " + myCar1.color + "이며, 속도는 " + myCar1.speed + "km입니다.");  
  
 myCar2.upSpeed(50)  
 *println*("자동차2의 색상은 " + myCar2.color + "이며, 속도는 " + myCar2.speed + "km입니다.");  
  
 myCar3.upSpeed(50)  
 *println*("자동차3의 색상은 " + myCar3.color + "이며, 속도는 " + myCar3.speed + "km입니다.");  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-9

[소스코드]

*/\* 예제3-9 Car.kt-생성자 추가 \*/  
/\* 02 생성자 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
  
 constructor(color: String, speed: Int) {  
 this.color = color  
 this.speed = speed  
 }  
  
 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  
 }  
}

예제3-10

[소스코드]

*/\* 예제3-10 exam06.kt-수정 \*/  
/\* 01 클래스 정의와 인스턴스 생성 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
  
 constructor(color: String, speed: Int) {  
 this.color = color  
 this.speed = speed  
 }  
  
 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)  
  
 myCar1.upSpeed(50)  
 *println*("자동차1의 색상은 " + myCar1.color + "이며, 속도는 " + myCar1.speed + "km입니다.");  
  
 myCar2.upSpeed(50)  
 *println*("자동차2의 색상은 " + myCar2.color + "이며, 속도는 " + myCar2.speed + "km입니다.");  
  
 myCar3.upSpeed(50)  
 *println*("자동차3의 색상은 " + myCar3.color + "이며, 속도는 " + myCar3.speed + "km입니다.");  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

예제3-11

[소스코드]

*/\* 예제3-11 Car.kt-메소드 오버로딩 추가 \*/  
/\* 03 메소드 오버로딩 \*/*class Car {  
 var color : String = ""  
 var speed : Int = 0  
  
 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  
 }  
}

예제3-12

[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  
 }  
}

[실행결과]

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

함수

exam11.kt

[소스코드]

*/\* exam11.kt \*/  
/\* 함수 \*/*fun sum(a:Int, b:Int) : Int {  
 var sum = a + b  
 return sum  
}  
fun sum1(a:Int, b:Int) : Int {  
 return a + b  
}  
fun sum2(a:Int, b:Int) : Int = a + b  
fun sum3(a:Int, b:Int) = a + b  
fun main() {  
 val result1 = *sum3*(10,20)  
 val result2 = *sum3*(60,70)  
 *println*(result1)  
 *println*(result2)  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

람다표현식

exam12.kt

[소스코드]

*/\* exam12.kt \*/  
/\* 람다표현식 \*/*fun main() {  
 val add = **{**x:Int, y:Int **->** x + y**}** val result3 = add(2,3)  
 val result4 = add(6,7)  
  
 *println*(result3)  
 *println*(result4)  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

exam13.kt

[소스코드]

*/\* exam13.kt \*/  
/\* 람다표현식 \*/*fun main() {  
 val mul = **{**x:Int, y:Int **->** x \* y**}** *println*(mul(2,3))  
 *println*(mul(4,5))  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

exam14.kt

[소스코드]

*/\* exam14.kt \*/  
/\* 람다표현식 \*/  
/\* (예제1) 인자가 없는 익명 함수 \*/*fun main() {  
 val greeting = fun(){*println*("Hello")}  
 val result = greeting()  
 *print*(result)  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명

exam15.kt

[소스코드]

*/\* exam15.kt \*/  
/\* 람다표현식 \*/  
/\* (예제2) 인자와 리턴값 있는 익명 함수 \*/*fun main() {  
 val greeting2 = **{**name: String, age: Int **->** "Hello. My name is $name. I am $age year old"**}** val result2 = greeting("dongyang", 20)  
 print(result2)  
}

[실행결과]

텍스트이(가) 표시된 사진

자동 생성된 설명