

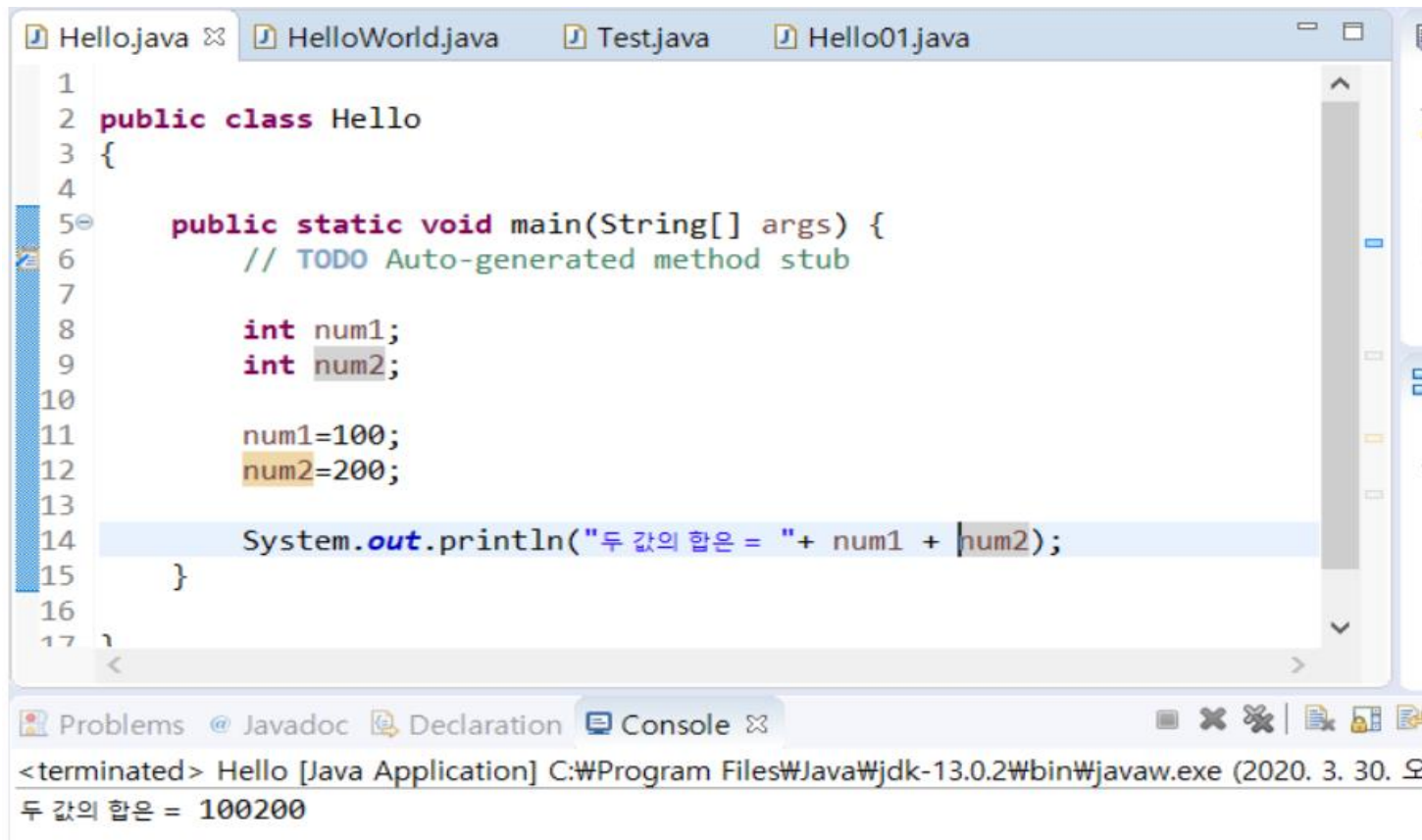
4주차-Java 프로그래밍

백석대학교 강윤희

(3주)Zoom 수업 진행

- 연산자 + 이해하기

System.out.println(7 + 1 + "Java " + 7 + 1);



The screenshot shows an IDE with a Java file named 'Hello.java'. The code defines a 'Hello' class with a 'main' method. Inside 'main', two integer variables 'num1' and 'num2' are declared and assigned the values 100 and 200 respectively. A 'System.out.println' statement prints the sum of 'num1' and 'num2' with the Korean text '두 값의 합은 = '. The console at the bottom shows the output: '두 값의 합은 = 100200'.

```
1
2 public class Hello
3 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         int num1;
9         int num2;
10
11         num1=100;
12         num2=200;
13
14         System.out.println("두 값의 합은 = " + num1 + num2);
15     }
16
17 }
```

<terminated> Hello [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 3. 30. 오후 10:30:30)
두 값의 합은 = 100200

3주 과제 진행 결과

```
shiftTest.java  nyurwa.java

1 package neka;
2 import java.util.Scanner;
3 public class shiftTest {
4     public static void main(String[] args) {
5         Scanner keyboard = new Scanner(System.in);
6         System.out.println("Enter an integer nuber");
7         int x = keyboard.nextInt();
8         System.out.println("Enter x for 2^x");
9         int y = keyboard.nextInt();
10        int myResult;
11        myResult = x << y;
12        System.out.println( x + " * " + "2^" + y + " = " + myResult);
13        System.out.println( "4 " + " * " + "2^" + "3" + " = " + ( 4 << 3));
14    }
15 }
16
17

1 import java.util.Scanner;
2 public class ShiftOperationTest {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7         Scanner keyboard = new Scanner(System.in);
8         System.out.println("Enter an integer nuber");
9         int x = keyboard.nextInt();
10        System.out.println("Enter x for 2^x");
11        int y = keyboard.nextInt();
12
13        int myResult;
14        myResult = x << y;
15
16        System.out.println( x + " * " + "2^" + y + " = " + myResult);
17        // System.out.println( "4 " + " * " + "2^" + "3" + " = " + ( 4 << 3));
18    }
19 }
20
21 }
```

Problems Javadoc Declaration Console

<terminated> shiftTest [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 30, 2020)

Enter an integer nuber
67
Enter x for 2^x
4
67 * 2^4 = 1072
4 * 2^3 = 32

수업 내용

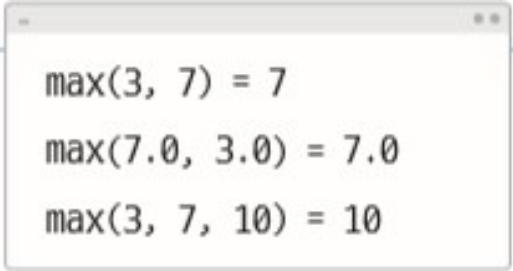
5	온라인	1. (3주차) 실습보조자료 Java 프로그래밍 Scanner	14분	예	모바일 미리보기 수정 삭제
6	온라인	2. 제어문과 메서드-조건문-동영상-01	18분	예	모바일 미리보기 수정 삭제
7	온라인	3. 제어문과 메서드-반복문-동영상-02	30분	예	모바일 미리보기 수정 삭제
8	온라인	4. 제어문과 메소드-메소드	26분	예	모바일 미리보기 수정 삭제
9	온라인	5. 객체 지향-소개-예습	12분	예	모바일 미리보기 수정 삭제

메서드

■ 메서드 오버로딩


- 메서드 시그니처(Method Signature) : 메서드 이름과 매개변수 개수, 데이터 타입, 순서를 의미
- 메서드 이름은 같지만 메서드 시그니처가 다른 메서드를 정의하는 것을 메서드 오버로딩(Method Overloading)이라고 함



```
1 public class OverloadDemo {  
2     public static void main(String[] args) {  
3         int i1 = 3, i2 = 7, i3 = 10;  
4         double d1 = 7.0, d2 = 3.0;  
5  
6         System.out.printf("max(%d, %d) = %d\n", i1, i2, max(i1, i2));  
7         System.out.printf("max(%.1f, %.1f) = %.1f\n", d1, d2, max(d1, d2));  
8         System.out.printf("max(%d, %d, %d) = %d\n", i1, i2, i3, max(i1, i2, i3));  
9     }  
10  
11     public static int max(int n1, int n2) {  
12         int result = n1 > n2 ? n1 : n2;  
13         return result;  
14     }  
15  
16     public static double max(double n1, double n2) {  
17         double result = n1 > n2 ? n1 : n2;  
18         return result;  
19     }  
20  
21     public static int max(int n1, int n2, int n3) {  
22         return max(max(n1, n2), n3);  
23     }  
24 }
```




```
max(3, 7) = 7  
max(7.0, 3.0) = 7.0  
max(3, 7, 10) = 10
```



과제 수행하기

 [일반 과제] 4주차-메소드 실습 [진행중]


 수정  삭제  과제평가




참고 자료 :  실습4주-메소드.pdf

3%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-03-30 00:00 ~ 2020-04-11 23:59	반영	공개	2020-04-18 00:00	 사용 	미허용	1명 / 27명 / 0명 / 28명



첨부된 내용을 참고하여 프로그램을 작성한 후 결과를 보고서로 제출하시오.

 [일반 과제] 4주차 제어문 [진행중]

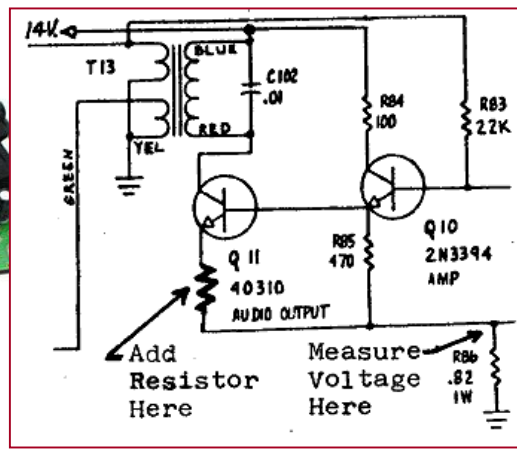
 수정  삭제  과제평가

참고 자료 :  03-제어문.pdf

3%

제출기간	성적반영	성적공개	성적공개일자	사용 여부	연장제출	제출자수 (제출 / 미제출 / 연장제출 / 수강인원)
2020-03-30 00:00 ~ 2020-04-11 23:59	반영	공개	2020-04-18 00:00	 사용 	미허용	1명 / 27명 / 0명 / 28명

프로그램내의 일련의 문장 수행을 반복하기 위한 제어문인 while과 for문을 사용하여 첨부된 1-6의 내용에 따라 Java 프로그램을 작성한다.



객체 지향

클래스와 객체(또는 클래스 인스턴스)



수업목표

- 객체란 무엇인지 이해한다
- 객체지향 프로그램과 절차지향 프로그램의 차이점을 이해한다
- 클래스와 객체의 관계를 이해한다
- 클래스를 작성하고 해당 클래스의 객체를 선언 및 생성한다

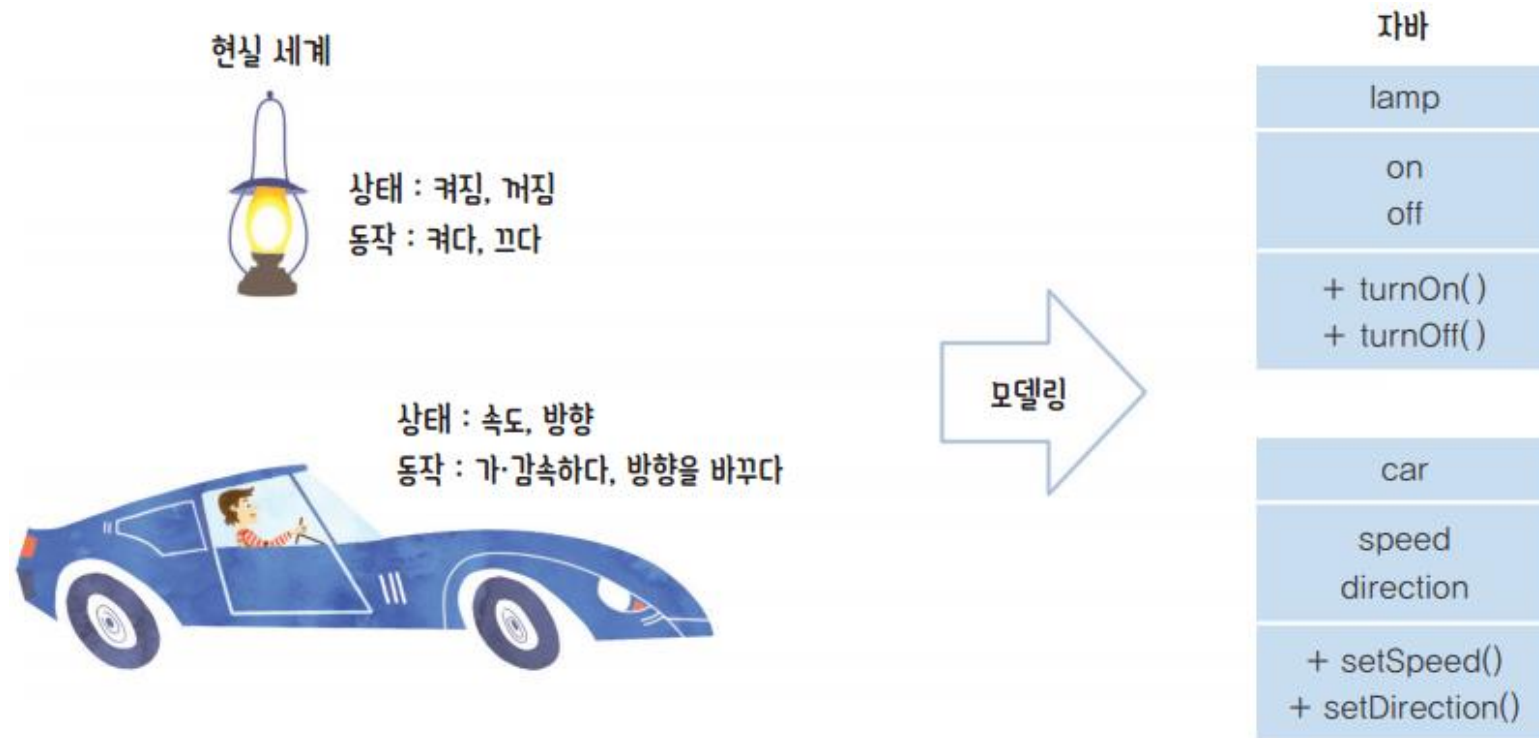
- 객체지향은 실세계에 존재하는 객체들을 중심으로 시스템을 구성하는 것
- 클래스/객체는 객체지향 프로그램의 기본적인 단위 모듈
- 객체지향의 개념과 클래스의 구조, 구성요소인 변수와 메소드 소개



객체 지향의 개요

■ 객체의 개념

- 소프트웨어 객체는 현실 세계의 객체를 필드와 메서드로 모델링한 것
- 소프트웨어 객체는 상태를 필드(Field)로 정의하고, 동작을 메서드(Method)로 정의
- 필드는 객체 내부에 선언된 변수를 의미하고, 메서드는 객체 내부에 정의된 동작



클래스 구성하기

- 클래스 찾아 공통적인 속성과 행동을 찾아본다



클래스 구성하기

■ 속성(필드) 와 동작(메소드)

- (문제)도형 편집하기 위한 프로그램을 하고자 한다.
 - 도형으로 생각하는 것을 삼각형, 사각형, 원 등이다.
 - 도형의 정보를 얻을 수 있는 기능을 포함하여야 한다.
 - 도형의 넓이를 구하는 기능을 포함하여야 한다.