과목: 객체지향프로그래밍및실습

교수: 최지웅 교수님

객체지향 프로그래밍 <실습 1>

03/30

홍지훈

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 4 | 6 | 8 | 10 | 12 |
| O | O | O | O | O | O |

이름: 홍지훈

학과: 소프트웨어학부

분반: 나

학번 : 20201777

2번문제

2자리의 정수 (10~99)를 입력 받아서 십의 자리와 일의 자리가 같은지 판별하는 프로그램.

2-1. 소스코드

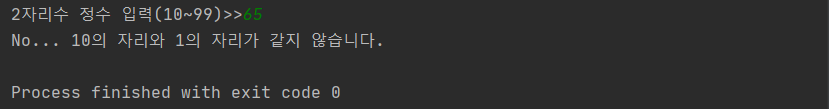
import java.util.Scanner;  
  
public class p2 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("2자리수 정수 입력(10~99)>>");  
 int num = scanner.nextInt();  
 if ((num / 10) == (num % 10)) {  
 System.*out*.println("Yes! 10의 자리와 1의 자리가 같습니다.");  
 } else {  
 System.*out*.println("No... 10의 자리와 1의 자리가 같지 않습니다.");  
 }  
 scanner.close();  
 }  
}

2-2. 결과

(1) 두 자리수가 같을 때 (77)



(2) 두 자리수가 다를 때 (65)



4번문제

정수 3개를 입력 받아서 중간크기의 값을 구하는 문제

4-1. 소스코드

import java.util.Scanner;  
  
public class p4 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("정수 3개 입력>>");  
 int n1 = scanner.nextInt();  
 int n2 = scanner.nextInt();  
 int n3 = scanner.nextInt();  
 int mid = n1;  
 if(((n2 >= n1) && (n2 <= n3)) || ((n2 <= n1) && n2 >= n3))  
 mid = n2;  
 else if(((n3 >= n1) && (n3 <= n2)) || ((n3 <= n1) && n3 >= n2))  
 mid = n3;  
  
 System.*out*.println("중간 값은 " + mid);  
 scanner.close();  
 }  
}

4-2. 결과

(1) 중간 값이 세번째에 있을 때 (20 100 33)



(2) 중간 값이 두번째에 있을 때 (10 20 30)



(2) 중간 값이 첫번째에 있을 때 (55 200 11)



(2) 2개의 정수가 같을 때 (35 72 35)



(2) 3개의 정수가 같을 때 (10 10 10)



6번문제

1~99 정수를 입력 받아서 369게임에서 박수를 시키는 문제

1개 있는 경우 (박수짝)

2개 있는 경우 (박수짝짝)

0개 있는 경우 (숫자만) (임의)

6-1. 소스코드

import java.util.Scanner;  
  
public class p6 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("1~99 사이의 정수를 입력하시오>>");  
 int n = scanner.nextInt();  
 int cnt = 0;  
 //십의자리수  
 if((n/10 == 3 || n/10 == 6 || n/10 == 9))  
 cnt++;  
 //일의자리수  
 if((n%10 == 3 || n%10 == 6 || n%10 == 9))  
 cnt++;  
   
 //결과 출력  
 if(cnt == 0)  
 System.*out*.println(n);  
 else if(cnt == 1)  
 System.*out*.println("박수짝");  
 else if(cnt == 2)  
 System.*out*.println("박수짝짝");  
   
   
 scanner.close();  
 }  
}

6-2. 결과

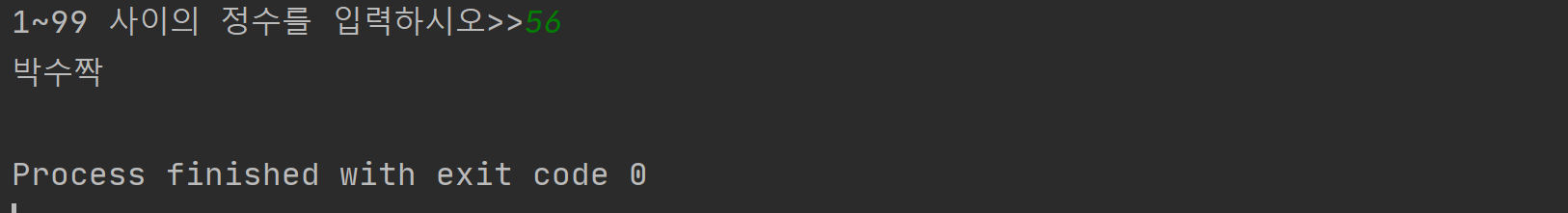
(1) 3의 배수가 0개 있는 경우, 1자리수(7)



(2) 3의 배수가1개 있는 경우, 1자리수(3)



(3) 3의 배수가 1개 있는 경우, 2자리수(56)



(4) 3의 배수가 2개 있는 경우(36)



8번문제

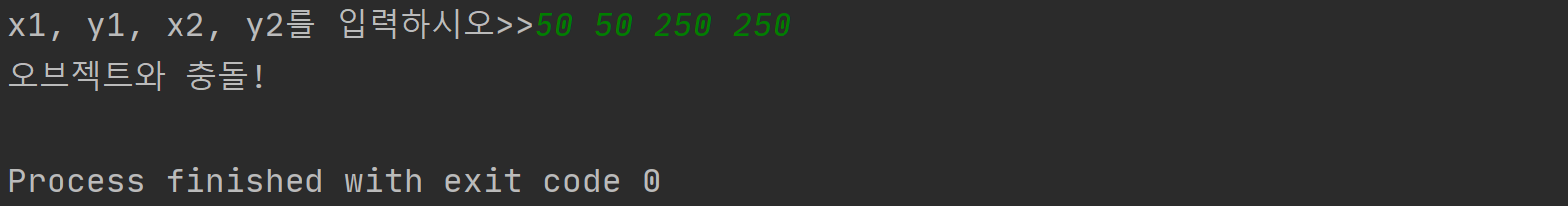
직사각형을 구성하는 두 점(x1, y1), (x2, y2)를 입력받아 (100, 100), (200, 200)의 두 점으로 이루어진 직사각형과 충돌하는지 판별하는 프로그램

8-1. 소스코드

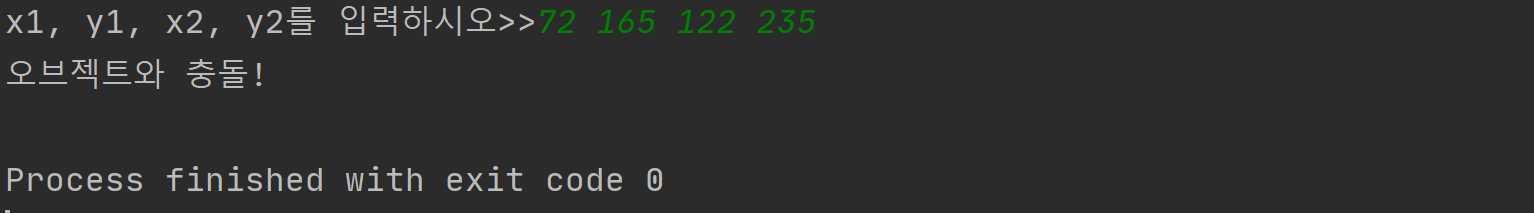
import java.util.Scanner;  
  
public class p8 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("x1, y1, x2, y2를 입력하시오>>");  
 int x1 = scanner.nextInt();  
 int y1 = scanner.nextInt();  
 int x2 = scanner.nextInt();  
 int y2 = scanner.nextInt();  
  
 if(((x1 <= 200) && (x2 >= 100)) && ((y1 <= 200) && (y2 >= 100)))  
 System.*out*.println("오브젝트와 충돌!");  
 else  
 System.*out*.println("오브젝트와 충돌하지 않음!");  
  
 scanner.close();  
 }  
}

8-2. 결과

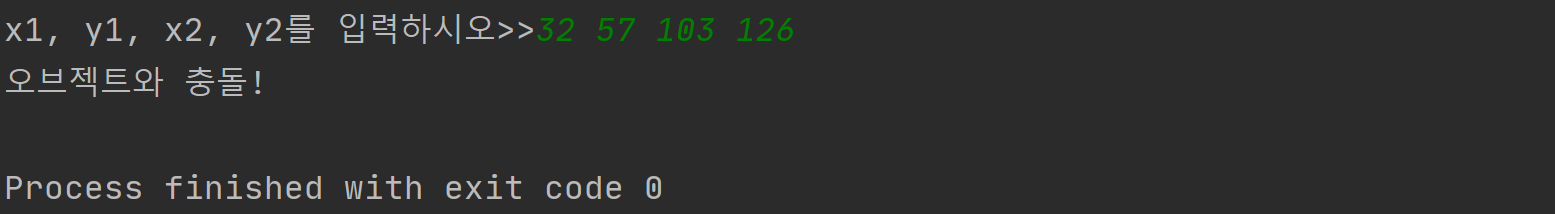
(1) 직사각형이 오른쪽 밑으로 겹칠 때 (50 50 250 250)



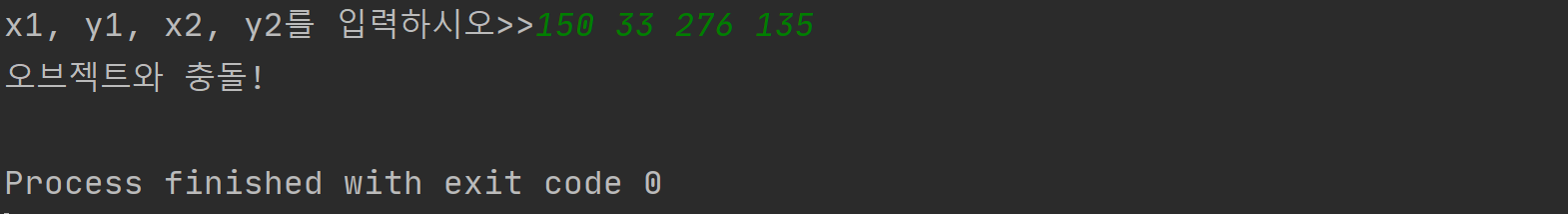
(2) 직사각형이 왼쪽 밑으로 겹칠 때 (72 165 122 235)



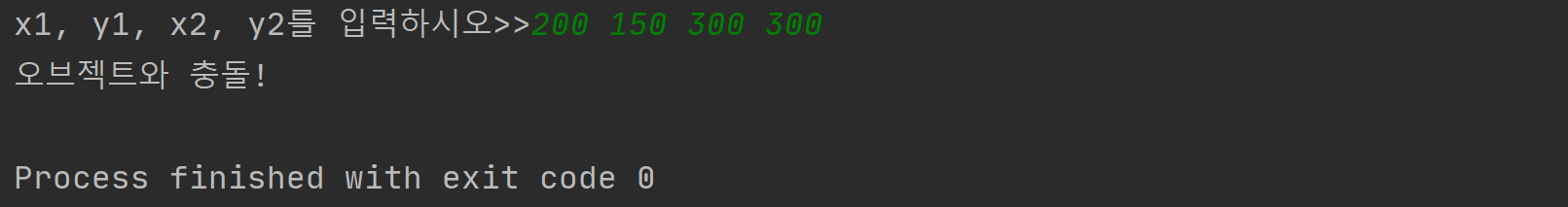
(3) 직사각형이 왼쪽 위로 겹칠 때 (32 57 103 126)



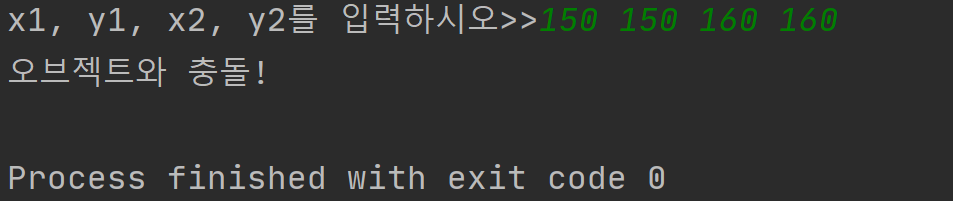
(4) 직사각형이 오른쪽 위로 겹칠 때 (150 33 276 135)



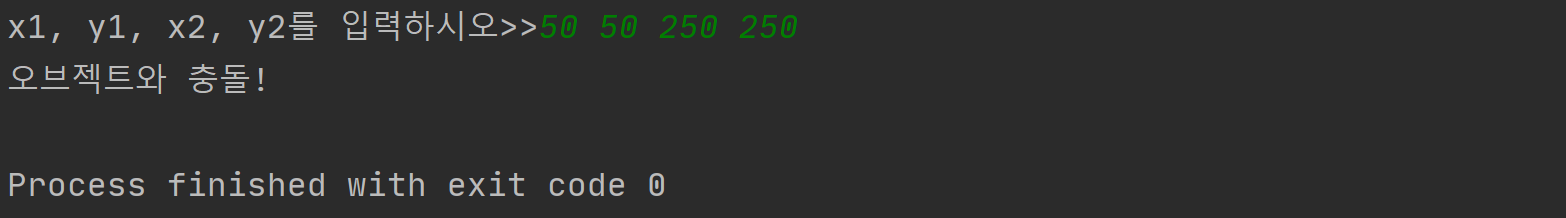
(5) 직사각형이 오른쪽으로 붙을 때 (200 150 300 300)



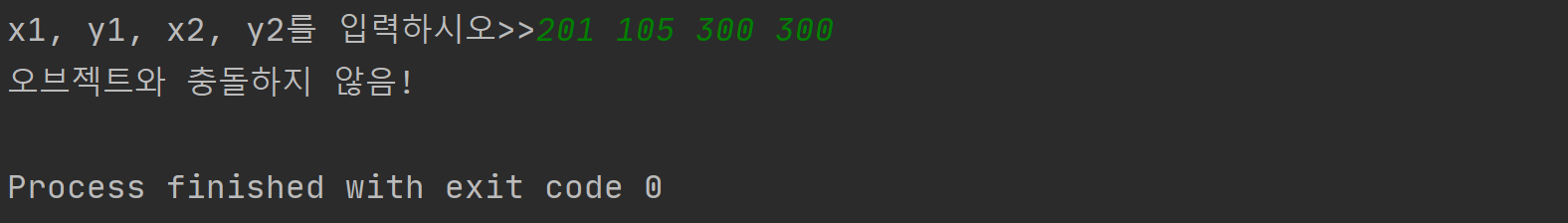
(6) 직사각형이 오브젝트 안쪽으로 들어갈 때 (150 150 160 160)



(7) 직사각형이 오브젝트 밖으로 감쌀 때 (50 50 250 250)



(7) 직사각형이 겹치지 않을 때 (201 105 300 300)



8-3. 특이사항

(1) 따로 예제가 없어서 x1, y1, x2, y2를 한번에 공백으로 받는 것을 가정했고 입력과 출력 문구를 임의로 지정함

(2) ‘충돌’이라는 단어를 사용하여 서로 변이 맞닿아 있는 상황?에서도 충돌로 인식하도록 만듬. 또한 두 오브젝트가 겹치는 모든 상황을 이야기함.

10번문제

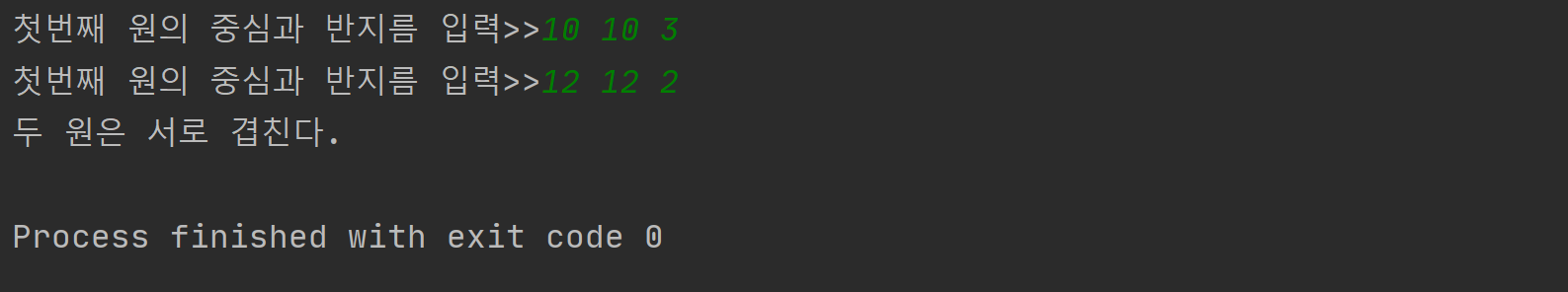
원의 중심 좌표(x, y)와 반지름을 2개 입력 받은 뒤 두 개의 원이 서로 겹치는지 판단

10-1. 소스코드

import java.util.Scanner;  
  
public class p10 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("첫번째 원의 중심과 반지름 입력>>");  
 int x1 = scanner.nextInt();  
 int y1 = scanner.nextInt();  
 int r1 = scanner.nextInt();  
 System.*out*.print("첫번째 원의 중심과 반지름 입력>>");  
 int x2 = scanner.nextInt();  
 int y2 = scanner.nextInt();  
 int r2 = scanner.nextInt();  
 int xdis = x2-x1;  
 if(xdis < 0) xdis \*= -1;  
 int ydis = y2-y1;  
 if(ydis < 0) ydis \*= -1;  
  
 int rdis = r1 + r2;  
  
 if(((xdis \* xdis) + (ydis \* ydis)) <= (rdis \* rdis))  
 System.*out*.println("두 원은 서로 겹친다.");  
 else  
 System.*out*.println("두 원은 서로 겹치지 않는다.");  
  
 scanner.close();  
 }  
}

10-2. 결과

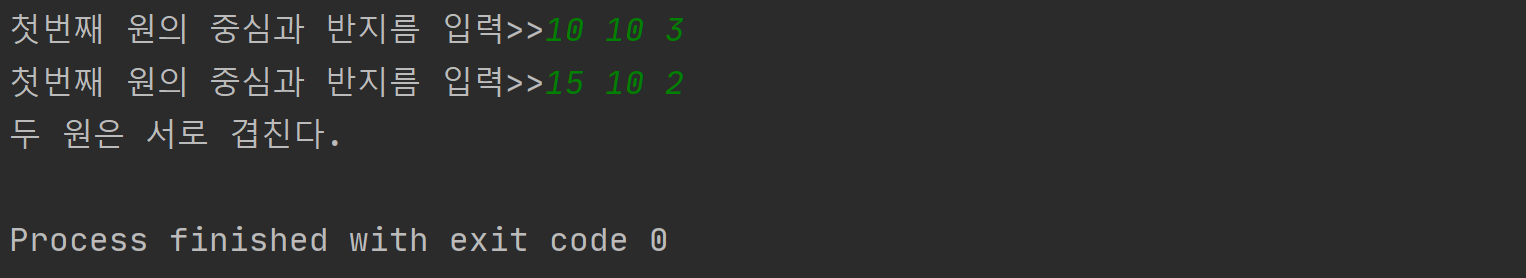
(1) 서로 겹칠 때 (10 10 3, 12 12 2)



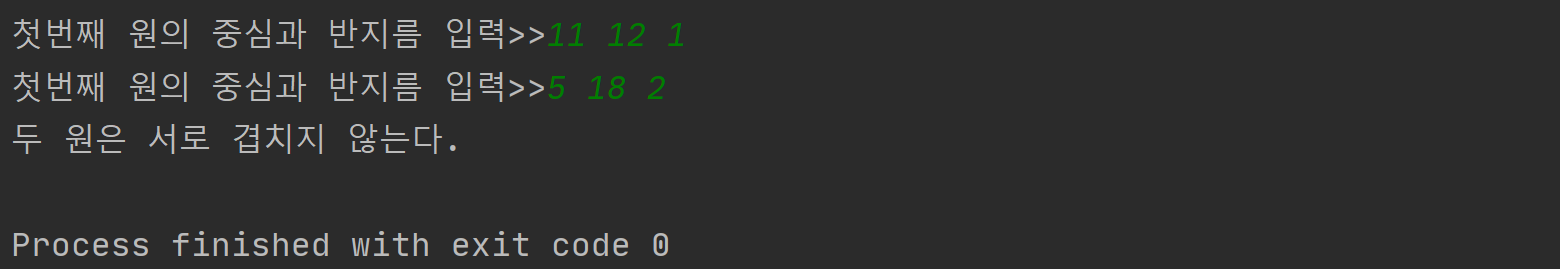
(2) 서로 겹칠 때, 두번째 원이 왼쪽 (15 14 5, 11 12 3)



(3) 두 원이 서로 맞다을 때 (10 10 3, 15 10 2)



(4) 서로 겹치지 않음 (11 12 1, 15 18 2)



10-3. 특이사항

겹처진다는 의미는 두 원이 맞닫는 것 또한 포함함.

12번문제

사칙 연산을 입력 받아 계산하는 프로그램

연산자는 +, -, \*, / 네가지, 피연산자는 모두 실수

0으로 나누기시 “0으로 나눌 수 없습니다.” 출력후 종료

(1) if-else 문으로 작성

(2) switch 문으로 작성

12-1. 소스코드

(1) if-else문

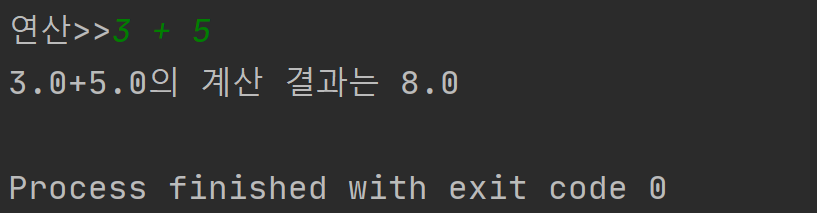
import java.util.Scanner;  
  
public class p12 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("연산>>");  
  
 double a = scanner.nextDouble();  
 String sign = scanner.next();  
 double b = scanner.nextDouble();  
  
 double result = 0;  
 boolean zero = false;  
 if(sign.equals("+"))  
 result = a + b;  
 else if(sign.equals("-"))  
 result = a - b;  
 else if(sign.equals("\*"))  
 result = a \* b;  
 else if(sign.equals("/")) {  
 if(b == 0)  
 zero = true;  
 else  
 result = a / b;  
 }  
  
  
 if(zero)  
 System.*out*.println("0으로 나눌 수 없습니다.");  
 else  
 System.*out*.println(a+sign+b+"의 계산 결과는 "+result);  
  
 scanner.close();  
 }  
}

(1) switch문

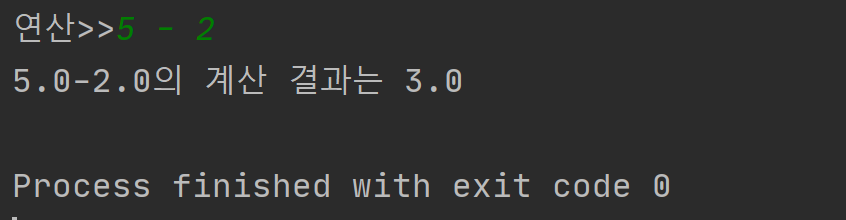
import java.util.Scanner;  
  
public class p12\_2 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("연산>>");  
  
 double a = scanner.nextDouble();  
 String sign = scanner.next();  
 double b = scanner.nextDouble();  
  
 double result = 0;  
 boolean zero = false;  
 switch(sign) {  
 case "+":  
 result = a + b;  
 break;  
 case "-":  
 result = a - b;  
 break;  
 case "\*":  
 result = a \* b;  
 break;  
 case "/":  
 if(b == 0)  
 zero = true;  
 else  
 result = a / b;  
 break;  
 default: break;  
 }  
  
  
 if(zero)  
 System.*out*.println("0으로 나눌 수 없습니다.");  
 else  
 System.*out*.println(a+sign+b+"의 계산 결과는 "+result);  
  
 scanner.close();  
 }  
}

12-2. 결과 (if문과 switch문의 결과값은 같습니다)

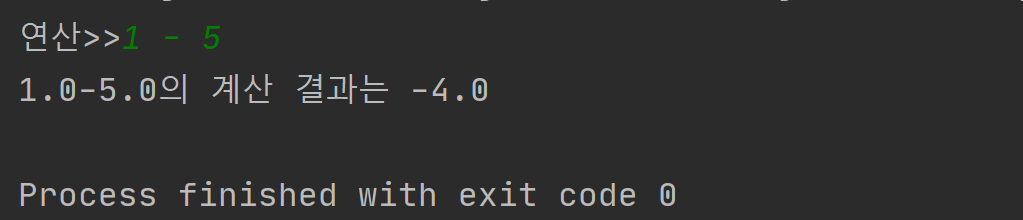
(1) 덧셈 (3 + 5)



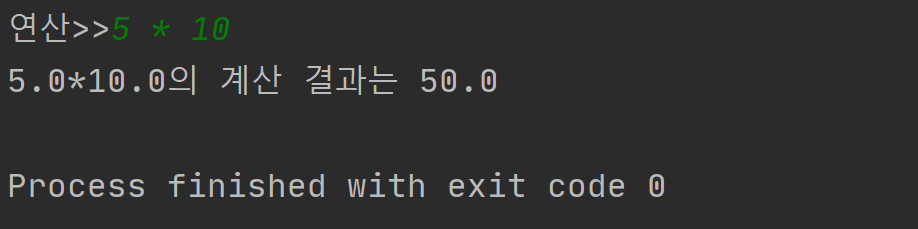
(2) 뺄셈 (5 – 2)



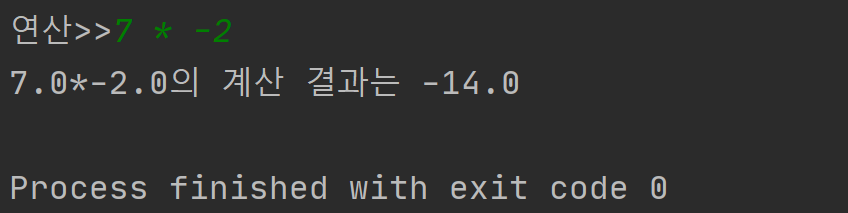
(3) 뺄셈(마이너스) (1 – 5)



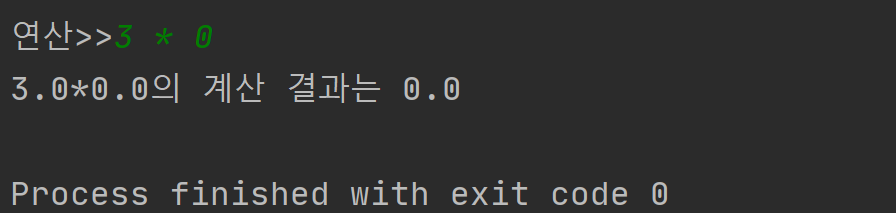
(4) 곱셈 (5 \* 10)



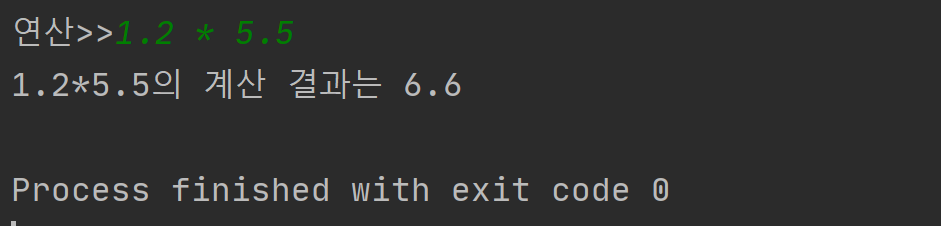
(5) 곱셈(마이너스) (7 \* -2)



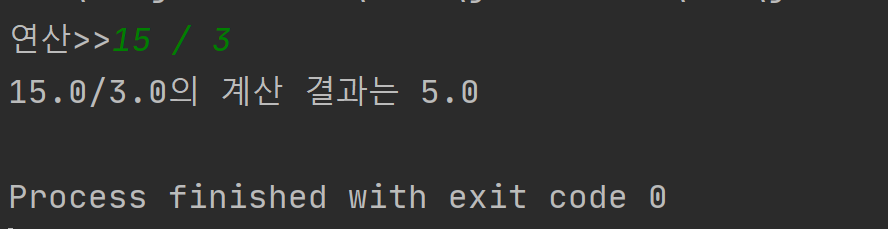
(6) 곱셈(0) (3 \* 0)



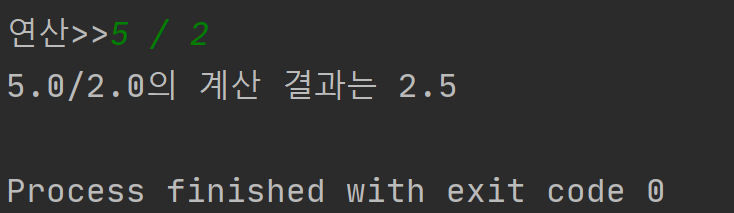
(7) 곱셈(소수점) (1.2 \* 5.5)



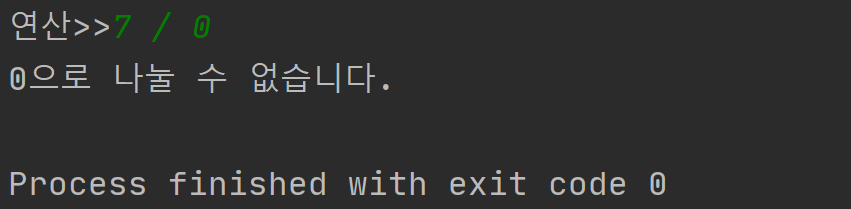
(8) 나눗셈 (15 / 3)



(9) 나눗셈(소수점) (5 / 2)



(10) 나눗셈(0) (7 / 0)



12-3. 특이사항

double형태로 println을 하였을 때 .0이 붙은 상태로 소수로 반환이 되는데 교수님께서 이렇게 해도 된다고 하셔서 그냥 이상태로 했습니다.