과목 : 객체지향 프로그래밍(가반)

교수 : 최지웅 교수

이름 : 김병준

학번 : 20162448

과제 #1

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

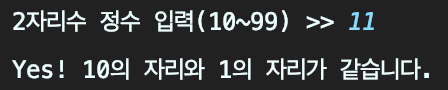
1. Solve2to2.java – 10의 자리와 1의 자리 비교 및 판별

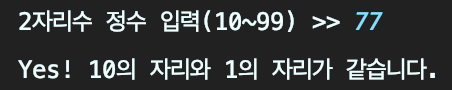
* Source

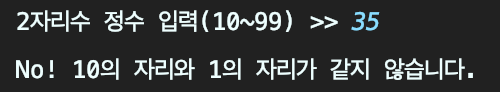
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Solve2to2 {  
 public static void main(String args[]) {  
 try {  
  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("2자리수 정수 입력(10~99) >> ");  
 int number = scanner.nextInt();  
 scanner.close();  
 if (number > 99 || number < 10) {  
 System.out.println("범위를 벗어난 입력입니다.");  
 main(args);  
 } else if (number / 10 == number % 10) {  
 System.out.println("Yes! 10의 자리와 1의 자리가 같습니다.");  
 } else {  
 System.out.println("No! 10의 자리와 1의 자리가 같지 않습니다.");  
 }  
 }catch (InputMismatchException e) {  
 System.out.println("정수를 입력해주세요.");  
 System.exit(0);  
 }  
 }  
}

* Result

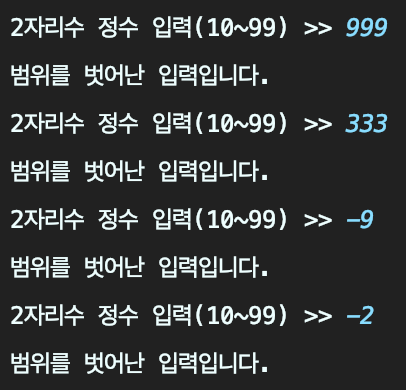
1. 9 < INPUT < 99



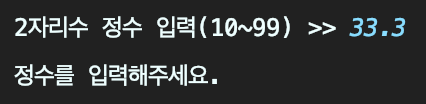




1. INPUT < 10 || INPUT > 99



1. INPUT != INTEGER



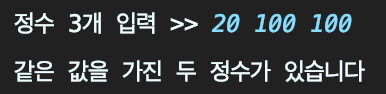
1. Solve2to4 – 정수 3개를 입력 받고 중간 값 출력

* Source

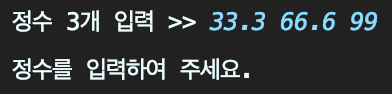
import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to4 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 int a = 0;  
 int b = 0;  
 int c = 0;  
 System.out.print("정수 3개 입력 >> ");  
 try {  
 a = scanner.nextInt();  
 b = scanner.nextInt();  
 c = scanner.nextInt();  
 } catch (InputMismatchException e) {  
 System.out.println("정수를 입력하여 주세요.");  
 System.exit(0);  
 }  
 scanner.close();  
 int min = 0, mid = 0, max = 0;  
 if (a == b && b == c) {  
 System.out.println("세 정수의 값이 모두 같습니다.");  
 System.exit(0);  
 } else if (a == b || b == c || a == c) {  
 System.out.println("같은 값을 가진 두 정수가 있습니다");  
 System.exit(0);  
 } else {  
 if (a > b) {  
 max = a;  
 if (b > c) { // a > b > c  
 mid = b;  
 min = c;  
 } else { // a > c > b  
 mid = c;  
 min = b;  
 }  
 } else if (b > c) {  
 max = b;  
 if (c > a) { // b > c > a  
 mid = c;  
 min = a;  
 } else { // b > a > c  
 mid = a;  
 min = c;  
 }  
 } else if (c > a) {  
 max = c;  
 if (a > b) { // c > a > b  
 mid = a;  
 min = b;  
 } else { //c > b > a  
 mid = b;  
 min = a;  
 }  
 }  
 }  
 System.out.println("중간값은 " + mid);  
 }  
}

* Result

1. INPUT = INTEGER, A != B != C



1. INPUT != INTEGER



1. A == B || B == C || A == C

개체이(가) 표시된 사진

자동 생성된 설명

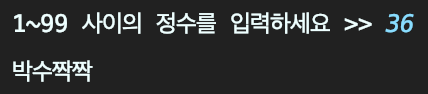
1. Solve2to6 – 369게임

* Source

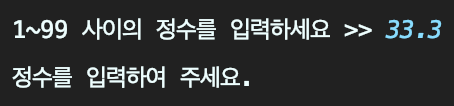
import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to6 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("1~99 사이의 정수를 입력하세요 >> ");  
 int num = 0;  
 try {  
 num = scanner.nextInt();  
 }catch (InputMismatchException e) {  
 System.out.println("정수를 입력하여 주세요.");  
 System.exit(0);  
 }  
 scanner.close();  
 int count = 0;  
 if (num < 0 || num > 99) {  
 System.out.println("입력한 값이 범위를 초과하였습니다.");  
 System.exit(0);  
 } else if (num < 10) {  
 if(num == 3 || num == 6 || num == 9) count++;  
 } else {  
 int tenposition = num / 10;  
 int oneposition = num % 10;  
 if (tenposition == 3 || tenposition == 6 || tenposition == 9) count++;  
 if (oneposition == 3 || oneposition == 6 || oneposition == 9) count++;  
 }  
 System.out.print("박수");  
 for(int i = 0; i < count; i++) {  
 System.out.print("짝");  
 }  
 }  
}

* Result

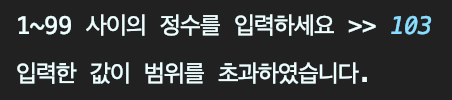
1. 0 < INPUT < 100



1. INPUT != INTEGER



1. INPUT < 1 || INPUT > 99



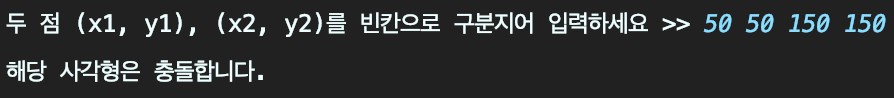
1. Solve2to8 – 두 직사각형의 충돌 판별

* Source

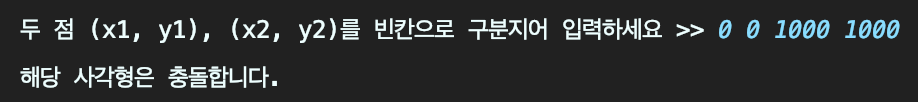
import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to8 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 Boolean check = false;  
 System.out.print("두 점 (x1, y1), (x2, y2)를 빈칸으로 구분지어 입력하세요 >> ");  
 int x1 = 0, y1 = 0, x2 = 0, y2 = 0;  
 try {  
 x1 = scanner.nextInt();  
 y1 = scanner.nextInt();  
 x2 = scanner.nextInt();  
 y2 = scanner.nextInt();  
 }catch (InputMismatchException e) {  
 System.out.println("정수를 입력하여 주세요.");  
 System.exit(0);  
 }  
 scanner.close();  
 if(x1 > x2 && y1 > y2) {  
 SwapLocation(x1, y1, x2, y2);  
 }  
 for (int i = x1; i <= x2; i++) {  
 if(check == true) break;  
 if(y1 > y2) {  
 for(int j = y2; j <= y1; j++) {  
 check =intRect(i, j, 100, 100, 200, 200);  
 if(check == true) break;  
 }  
 } else {  
 for (int j = y1; j <= y2; j++) {  
 check = intRect(i, j, 100, 100, 200, 200);  
 if (check == true) break;  
 }  
 }  
 }  
 if(check == null) {  
 System.out.println("검증이 이루어지지 않았습니다.");  
 } else if (check == true){  
 System.out.println("해당 사각형은 충돌합니다.");  
 } else {  
 System.out.println("해당 사각형은 충돌하지 않습니다.");  
 }  
 }  
  
 public static boolean intRect(int x, int y, int rectx1, int recty1, int rectx2, int recty2) {  
 if ((x >= rectx1 && x <= rectx2) && (y >= recty1 && y <= recty2))  
 return true;  
 return false;  
 }  
 public static void SwapLocation(int x1, int y1, int x2, int y2) {  
 int temp = 0;  
 temp = x1;  
 x1 = x2;  
 x2 = temp;  
  
 temp = y1;  
 y1 = y2;  
 y2 = temp;  
 }  
}

* Result

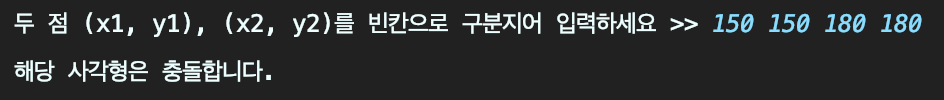
1. 기존의 사각형과 충돌할 경우



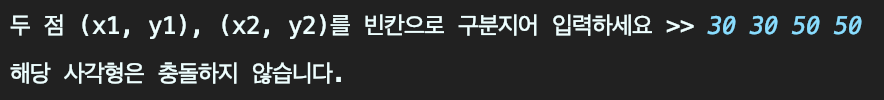
1. INPUT이 기존의 사각형을 포함하는 경우



1. 기존의 사각형이 INPUT을 포함하는 경우



1. INPUT과 기존의 사각형이 서로 떨어져있는 경우



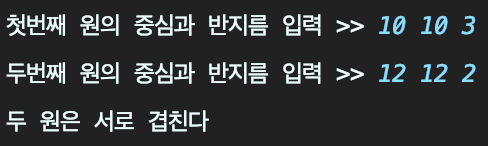
1. Solve2to10 – 두 원의 충돌 판별

* Source

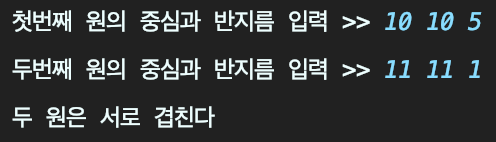
import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to10 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 int x1 = 0, y1 = 0, radius1 = 0, x2 = 0, y2 = 0, radius2 = 0;  
 try {  
 System.out.print("첫번째 원의 중심과 반지름 입력 >> ");  
 x1 = scanner.nextInt();  
 y1 = scanner.nextInt();  
 radius1 = scanner.nextInt();  
 System.out.print("두번째 원의 중심과 반지름 입력 >> ");  
 x2 = scanner.nextInt();  
 y2 = scanner.nextInt();  
 radius2 = scanner.nextInt();  
 }catch (InputMismatchException e) {  
 System.out.println("정수를 입력하여 주세요.");  
 System.exit(0);  
 }  
 scanner.close();  
 if (x1 > x2 && y1 > y2) {  
 SwapLocation(x1, y1, radius1, x2, y2, radius2);  
 }  
 if(CheckCircleCrash(x1, y1, radius1, x2, y2, radius2) == true) {  
 System.out.print("두 원은 서로 겹친다");  
 } else {  
 System.out.print("두 원은 서로 겹치지 않는다.");  
 }  
 }  
 public static boolean CheckCircleCrash(int x1, int y1, int radius1, int x2, int y2, int radius2) {  
 if(Math.sqrt(Math.pow((x2 - x1),2) + Math.pow((y2 - y1),2)) > radius1 + radius2) {  
 return false;  
 }  
 return true;  
 }  
 public static void SwapLocation(int x1, int y1, int radius1, int x2, int y2, int radius2) {  
 int temp = 0;  
 temp = x1;  
 x1 = x2;  
 x2 = temp;  
 temp = y1;  
 y1 = y2;  
 y2 = temp;  
 temp = radius1;  
 radius1 = radius2;  
 radius2 = temp;  
 }  
}

* Result

1. 두 원이 충돌하는 경우



1. 한 원이 다른 한원을 포함하는 경우



1. 두 원이 서로 충돌하지 않는 경우

실외이(가) 표시된 사진

자동 생성된 설명

1. Solve2to12 – 사칙연산 계산 프로그램

* Source – Solve2to12A : if-else문

import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to12 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 double result = 0;  
 double num1 = 0, num2 = 0;  
 String op = null;  
 char operator = 0;  
 System.out.print("연산 >> ");  
 try {  
 num1 = scanner.nextDouble();  
 op = scanner.next();  
 num2 = scanner.nextDouble();  
 operator = op.charAt(0);  
 } catch (InputMismatchException e) {  
 System.out.println("수식이 잘못되었습니다.");  
 System.exit();  
 }  
 scanner.close();  
 if (operator == '+') result = num1 + num2;  
 else if (operator == '-') result = num1 - num2;  
 else if (operator == '\*') result = num1 \* num2;  
 else if (operator == '/') {  
 // double형은 0으로 나누면 Arithmetic Exception 대신 infinity가 도출.  
 // 따라서 예외처리를 try-catch 대신 조건문 활용  
 if (num2 == 0) {  
 System.out.println("0으로 나눌 수 없습니다.");  
 System.exit(0);  
 } else result = num1 / num2;  
  
 } else System.out.println("잘못된 연산자입니다.");  
  
  
 System.out.println(num1 + "" + operator + "" + num2 + "의 계산결과는 " + result);  
 }  
}

* Source – Solve2to12B : switch문

import java.util.InputMismatchException;  
import java.util.Scanner;  
public class Solve2to12 {  
 public static void main(String args[]) {  
 Scanner scanner = new Scanner(System.in);  
 double result = 0;  
 double num1 = 0, num2 = 0;  
 String op = null;  
 char operator = 0;  
 System.out.print("연산 >> ");  
 try {  
 num1 = scanner.nextDouble();  
 op = scanner.next();  
 num2 = scanner.nextDouble();  
 operator = op.charAt(0);  
 } catch (InputMismatchException e) {  
 System.out.println("수식이 잘못되었습니다.");  
 System.exit(0);  
 }  
 scanner.close();  
 switch (operator) {  
 default:  
 System.out.println("잘못된 연산자입니다.");  
 System.exit(0);  
 case '+':  
 result = num1 + num2;  
 break;  
 case '-':  
 result = num1 - num2;  
 break;  
 case '\*':  
 result = num1 \* num2;  
 break;  
 case '/':  
 // double형은 0으로 나누면 Arithmetic Exception 대신 infinity가 도출.  
 // 따라서 예외처리를 try-catch 대신 조건문 활용  
 if (num2 == 0) {

System.out.println("0으로 나눌 수 없습니다.");

System.exit(0);

}  
 else result = num1 / num2;  
 break;  
 }  
 System.out.println(num1 + "" + operator + "" + num2 + "의 계산결과는 " + result);  
 }  
}

* Result

1. 정상적인 수식

개체이(가) 표시된 사진

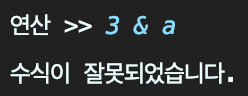
자동 생성된 설명

1. 0으로 나누기를 하였을 때

개체이(가) 표시된 사진

자동 생성된 설명

1. 잘못된 수식



개체이(가) 표시된 사진

자동 생성된 설명

개체이(가) 표시된 사진

자동 생성된 설명