

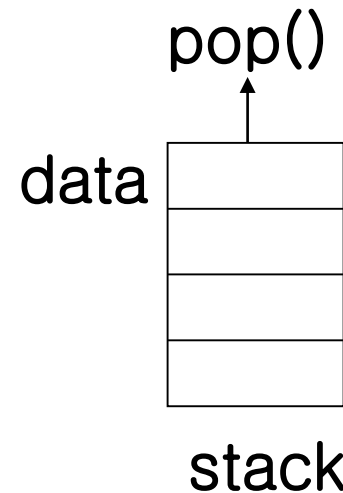
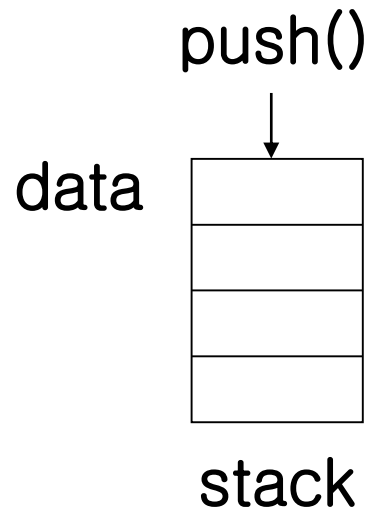
JAVA 프로그램 실습

Class 사용하기

경북대학교
소프트웨어융합과
배희호 교수

Stack 구현

- Stack은 Data를 저장하는 배열, Data를 저장하는 push() 메소드, 저장된 Data를 반환하는 pop() 메소드를 지원함
 - 즉 Stack을 정의하기 위해서는 내부 Data(배열)와 이를 조작하는 메소드를 함께 정의해야 함
- Stack은 가장 나중에 저장된 Data를 반환하기 때문에 LIFO(Last In First Out)라고 불리기도 함



Stack 구현

```
public class Stack {  
    private int[] data;  
    private int ptr;  
  
    public Stack() {  
        data = new int[100];  
        ptr = 0;  
    }  
    void push(int x) {  
        data[ptr] = x;  
        ptr++;  
    }  
    int pop() {  
        ptr--;  
        return(data[ptr]);  
    }  
}
```

변수 ptr은 다음 데이터를 저장할
배열의 첨자(index)를 저장하고 있음

Program이 실행될 때에는 Stack에는
아무런 Data가 저장되어
있지 않으므로,
ptr의 값은 0으로 초기화 됨

Stack 구현

```
public static void main( String args[] ) {  
    Stack stack = new Stack();  
    int i;  
  
    stack.push(4);  
    stack.push(7);  
  
    i = stack.pop();  
    System.out.println("result = " + i);  
    stack.push(12);  
    i = stack.pop();  
    System.out.println("result = " + i);  
    i = stack.pop();  
    System.out.println("result = " + i);  
}
```

```
result = 7  
result = 12  
result = 4
```

최대, 최소값(I)

- 3개의 정수를 전달받아, 최대값과 최소값을 구하는 메소드를 만들어보자.

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
  
    System.out.print("정수 3개를 입력 : ");  
    int num1 = keyboard.nextInt();  
    int num2 = keyboard.nextInt();  
    int num3 = keyboard.nextInt();  
  
    MaxMin.showInfo(num1, num2, num3);  
}
```

최대, 최소값(I)

```
public class MaxMin {  
    static int max(int num1, int num2, int num3) {  
        int max = num1;  
  
        // 방법1  
        if (max < num2) max = num2;  
        if (max < num3) max = num3;  
  
        // 방법2  
        max = max < num2 ? (num2 < num3 ? num3 : num2) :  
            (max < num3 ? num3 : max);  
  
        return max;  
    }  
}
```

최대, 최소값(1)

```
static int min(int num1, int num2, int num3) {  
    int min = num1;  
    if (min > num2) min = num2;  
    if (min > num3) min = num3;  
  
    min = min > num2 ? (num2 > num3 ? num3 : num2) :  
            (min > num3 ? num3 : min);  
  
    return min;  
}  
  
static void showInfo(int num1, int num2, int num3) {  
    System.out.println("최대값: " + max(num1, num2, num3));  
    System.out.println("최소값: " + min(num1, num2, num3));  
}  
}
```

최대, 최소값(II)

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
  
    System.out.print("정수 3개를 입력 : ");  
    int num1 = keyboard.nextInt();  
    int num2 = keyboard.nextInt();  
    int num3 = keyboard.nextInt();  
  
    MaxMin maxMin = new MaxMin(num1, num2, num3);  
    System.out.println(maxMin);  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

최대, 최소값(II)

```
public class MaxMin {  
    private int num1;  
    private int num2;  
    private int num3;  
  
    public MaxMin(int num1, int num2, int num3) {  
        this.num1 = num1;  
        this.num2 = num2;  
        this.num3 = num3;  
    }  
  
    public int max() {  
        int max = num1;  
        if (max < num2) max = num2;  
        if (max < num3) max = num3;  
        // 방법2  
        // max = max < num2 ? (num2 < num3 ? num3 : num2) :  
        //                                     (max < num3 ? num3 : max);  
        return max;  
    }  
}
```

최대, 최소값(II)

```
public int min() {  
    int min = num1;  
    if (min > num2) min = num2;  
    if (min > num3) min = num3;  
  
    min = min > num2 ? (num2 > num3 ? num3 : num2) :  
        (min > num3 ? num3 : min);  
    return min;  
}
```

@Override

```
public String toString() {  
    String result = "";  
    result += "num1 = " + num1 + ", num2 = " + num2 +  
        ", num3 = " + num3;  
    result += "\n최대값: " + max();  
    result += "\n최소값: " + min();  
    return result;  
}
```

정수 더하기

- 정수 2개를 전달받아(인자, argument) 2수를 더한 결과를 return해주는 sum() 메소드를 포함한 클래스 Adder를 사용하여라.

씨호랑이 수학

연산 복습
천안 죽곡 10) 천안 죽곡 10) 천안 죽곡 10) 천안 죽곡 10)

이름: _____

☺ 맞선을 하세요.

$1+3=$ <input type="text"/>	$2+5=$ <input type="text"/>
$6+2=$ <input type="text"/>	$8+1=$ <input type="text"/>
$4+4=$ <input type="text"/>	$5+1=$ <input type="text"/>
$3+2=$ <input type="text"/>	$2+5=$ <input type="text"/>
$7+2=$ <input type="text"/>	$4+1=$ <input type="text"/>

천안 죽곡 10) <http://www.coengyikid.com> / Tel. 070-7579-2080

정수 더하기

```
public class Adder {  
    private int num1;  
    private int num2;  
  
    public Adder(int num1, int num2) {  
        this.num1 = num1;  
        this.num2 = num2;  
    }  
    public int sum() {  
        return num1 + num2;  
    }  
    @Override  
    public String toString() {  
        String result;  
        result = "num1 = " + num1 + ", num2 = " + num2;  
        result += String.format("Wn %d + %d = %d", num1, num2, sum());  
        return result;  
    }  
}
```

정수 더하기

■ Main 클래스

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    System.out.print("정수 2개를 입력 : ");  
    int num1 = keyboard.nextInt();  
    int num2 = keyboard.nextInt();  
  
    Adder adder = new Adder(num1, num2);  
    System.out.printf("%d + %d = %d\n", num1, num2, num1 + num2);  
    System.out.println(adder);  
}
```

안전한 배열

- 만약 index가 배열의 크기를 벗어나게 되면 실행 Error 발생한다. 따라서 실행 Error를 발생하지 않는 안전한 배열을 작성하여 보자.

SafeArray
-a[] : int -length : int
+get(int index) : int +put(int index, int value) : void

안전한 배열

■ SafeArray.JAVA

```
public class SafeArray {  
    private int[] array;  
    public int length;  
  
    public SafeArray(int size) {  
        array = new int[size];  
        length = size;  
    }  
  
    public int get(int index) {  
        if (index >= 0 && index < length) {  
            return array[index];  
        }  
        return -1;  
    }  
}
```

안전한 배열

■ SafeArray.JAVA

```
public void put(int index, int value) {  
    if (index >= 0 && index < length) {  
        array[index] = value;  
    } else  
        System.out.println("잘못된 인덱스 " + index);  
}  
}
```


안전한 배열

■ Main.JAVA

```
public static void main(String[] args) {  
    SafeArray array = new SafeArray(3);  
  
    for (int i = 0; i < (array.length + 1); i++) {  
        array.put(i, i * 10);  
    }  
}
```

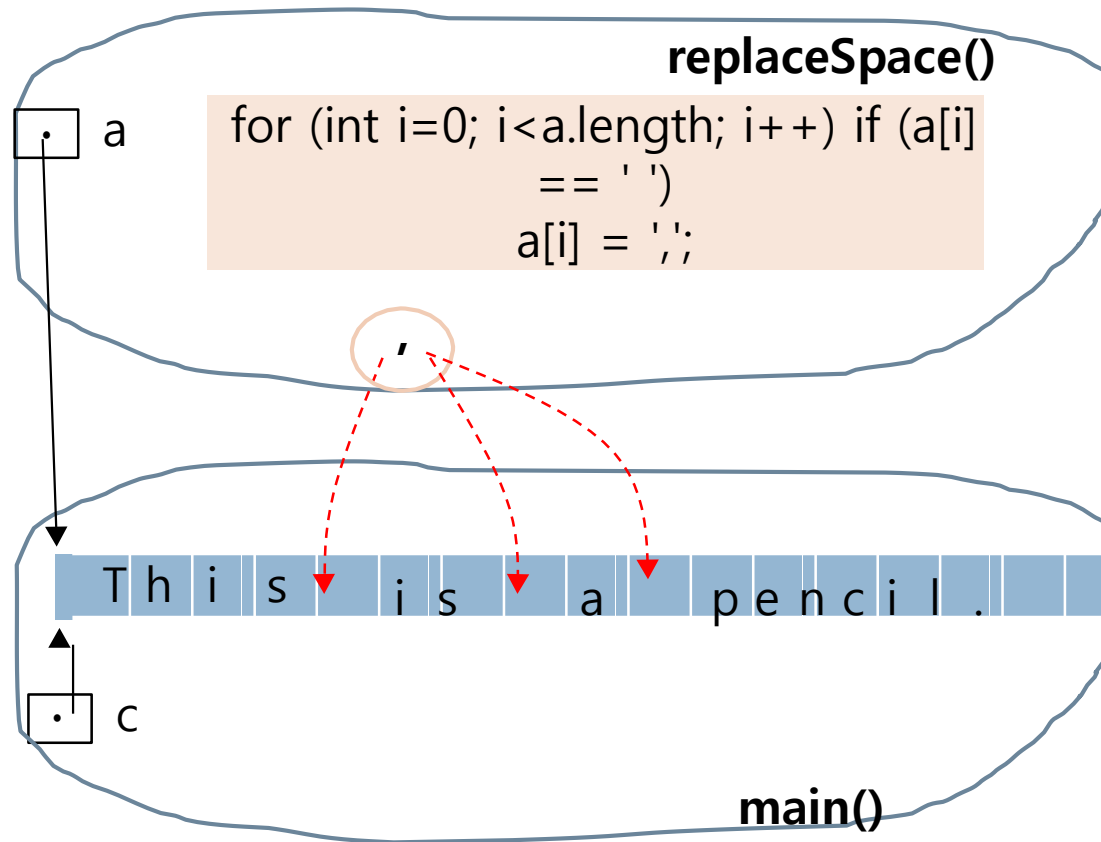


Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

문자 변환

- char[] 배열을 전달받아 배열 속의 공백(' ') 문자를 ','로 대체하는 메소드를 작성하라.



문자 변환

- char[] 배열을 전달받아 출력하는 printWord() 메소드와 배열 속의 공백(' ') 문자를 ',' 로 대체하는 replace() 메소드를 작성해보자.

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    String msg;

    System.out.print("문장을 입력하세요: ");
    msg = keyboard.nextLine();
    char[] arr = new char[msg.length()];

    for (int i = 0; i < arr.length; i++) {
        arr[i] = msg.charAt(i);
    }

    Word.showInfo(arr);
}
```

문자 변환

```
public class Word {  
    static void printWord(char[] arr) {  
        // 문자형 배열은 바로 그 값을 나타낸다.  
        // System.out.println(" " + arr); // 주소값  
        System.out.println(arr);  
    }  
  
    static void replace(char[] arr) {  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] == ' ') {  
                arr[i] = ',';  
            }  
        }  
        System.out.println(arr);  
    }  
}
```

문자 변환

```
static void showInfo(char[] arr) {  
    System.out.print("변경전: ");  
    printWord(arr);  
    System.out.print("변경후: ");  
    replace(arr);  
}  
}
```

이름을 전달

- 학번을 전달하면, 학생의 정보를 출력해주는 메소드 `showInfo()`를 만들어보자

이름을 전달

- 이름(문자) 1개를 전달받아 "나는 이름 입니다." 라는 출력문을 보여주는 인스턴스 showName()을 만들어라.

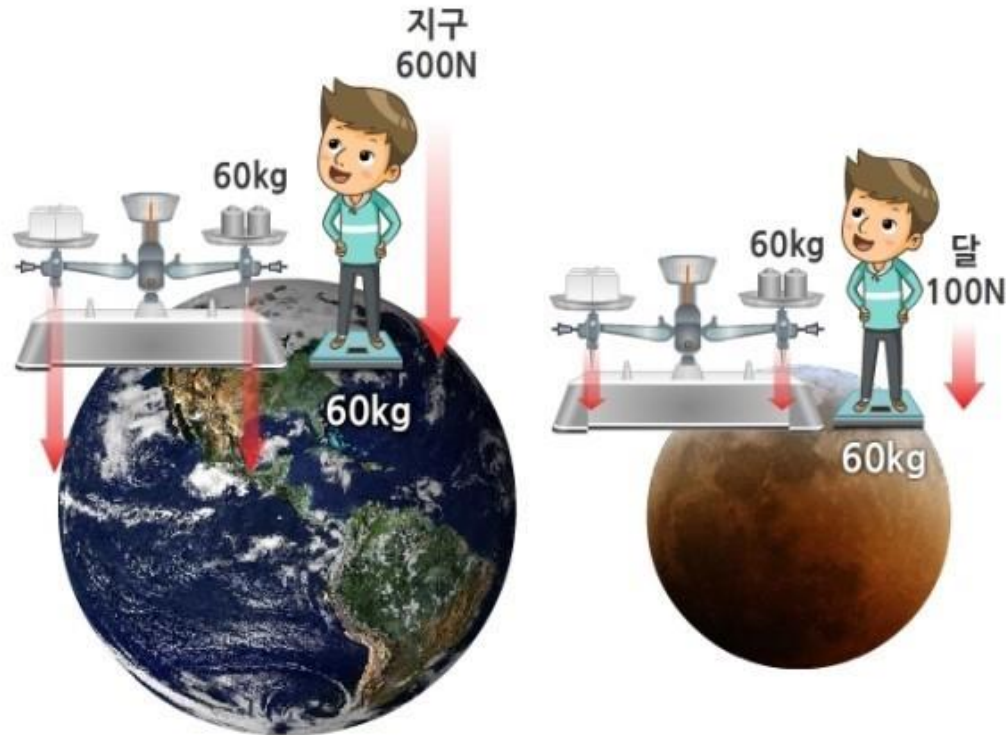
```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    String name;  
  
    System.out.print("이름을 입력해주세요: ");  
    name = keyboard.nextLine();  
  
    ShowName myName = new ShowName();  
    myName.showName(name);  
}
```

이름을 전달

```
public class ShowName {  
    private String name;  
  
    void showName(String name) {  
        System.out.println("나는 " + name + "입니다.");  
    }  
}
```


달에서의 몸무게

- 당신의 몸무게(weight)를 입력 받아서 달(Moon)에서의 몸무게로 환산하는 Program을 작성하여라.



- 달의 중력은 지구의 16.5%밖에는 되지 않음

달에서의 몸무게

- Data와 Information 파악하기
 - Data는 지구에서의 몸무게 (명사) – Member 변수
 - Information는 달에서의 몸무게 (동사) – Method
- Moon Class 구현

Moon
-earthWeight : int
+moonWeight() : float +toString() : String

달에서의 몸무게

■ 실행 결과

당신의 몸무게 입력 : 65

나의 몸무게 = 65 Kg, 달에서의 몸무게 = 10.73 Kg

달에서의 몸무게

■ Moon.JAVA

```
public class Moon {  
    private int earthWeight;  
  
    public Moon() {  
    }  
  
    public Moon(int earthWeight) {  
        this.earthWeight = earthWeight;  
    }  
    public float moonWeight() {  
        return earthWeight * (16.5f / 100);  
    }  
    @Override  
    public String toString() {  
        return "나의 몸무게 = " + earthWeight + " Kg, " +  
            "달에서의 몸무게 = " + String.format("%.2f Kg", moonWeight());  
    }  
}
```

달에서의 몸무게

■ Main.JAVA

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
  
    System.out.print("당신의 몸무게 입력 : ");  
    int weight = keyboard.nextInt();  
  
    Moon moon = new Moon(weight);  
    System.out.println(moon);  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

담배의 수명 단축

- 담배 1개피를 피우면 수명이 7분 단축 된다고 합니다. 평생 담배를 피우면 얼마의 시간이 수명 단축 될까요 ?
 - 단 1년은 365일로 한다.



당신은 하루에 몇 개피의 담배를 피우는가? 20

당신은 몇년동안 피우셨나요? 30

담배를 하루에 20개피씩 30년을 피우면,
단축되는 수명은 2년 334일 14시간 0분

담배의 수명 단축

■ Tabaco Class 설계

Tabaco
-year : int (몇 년간 피우셨나요) -cigar : int (하루에 몇 개피씩 피우나요)
+setYear(int) : void +setCigar(int) : void +calcSmoking() : int (피운 담배 개수) +calcLife() : int (수명 단축 시간) +calcDay() : int (년,월,일로 표현) +toString() : String

담배의 수명 단축

■ Tabaco.JAVA

```
public class Tabaco {  
    private int year;  
    private int cigar;  
  
    public Tabaco(int year, int cigar) {  
        this.year = year;  
        this.cigar = cigar;  
    }  
  
    private int calcSmoking() {  
        final int oneYear = 365;  
        return year * oneYear;  
    }  
  
    private int calcLife() {  
        final int time = 7;  
        return cigar * calcSmoking() * time;  
    }  
}
```


담배의 수명 단축

■ Tabaco.JAVA

```
private String calcDay() {  
    int minute = calcLife() % 60;  
    int hour = calcLife() / 60;  
    int day = hour / 24;  
    hour %= 24;  
    int year = day / 365;  
    day %= 365;  
  
    return String.format("%d년 %d일 %d시간 %d분", year, day, hour, minute);  
}
```

@Override

```
public String toString() {  
    return "담배를 하루에 " + cigar + "개피씩 " + year + "년을 피우면, " +  
        "Wn 단축되는 수명은 " + calcDay();  
}  
}
```

담배의 수명 단축

■ Main.JAVA

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
  
    System.out.print("당신은 하루에 몇 개피의 담배를 피우는가? ");  
    int cigar = keyboard.nextInt();  
    System.out.print("당신은 몇년동안 피우셨나요? ");  
    int year = keyboard.nextInt();  
  
    Tabaco tabaco = new Tabaco(year, cigar);  
    System.out.println(tabaco);  
}
```

Vending Machine

- Vending Machine에서 동전으로 거스름돈을 지불하는 시스템을 만들어보자.

물건의 가격 : 2300

지불액 : 2500

-----동전의 개수-----

거스름돈 : 200 원

500 원 동전의 수 : 0 개

100 원 동전의 수 : 2 개

50 원 동전의 수 : 0 개

10 원 동전의 수 : 0 개

5 원 동전의 수 : 0 개

1 원 동전의 수 : 0 개

Vending Machine

■ 문제 분석

- 물건 값 : 정수만 허용
- 지급한 현금 : 정수만 허용 (예: `int pay = 3900;`)
- 사용 가능한 동전 : 500원/100원/50원/10원/5원/1원
- 물건의 가격과 지불한 현금은 임의대로 주어졌을 때, 거스름돈을 동전의 종류별로 몇 개씩 주어야 하는지 산출하세요
- 단, 동전의 수가 가장 적게 지불하세요

■ Hint

- `/`(divide operator), `%`(modular operator) 연산자를 이용하여 동전의 개수를 구함
- 이때, 금액이 큰 금액부터 순차적으로 계산

Vending Machine

■ Change.JAVA

```
public class Change {  
    private int money;  
  
    public Change(int money) {  
        this.money = money;  
    }  
    private String change() {  
        String result = "";  
        int[] coinList = {500, 100, 50, 10, 5, 1};  
        int coin;  
  
        for(int i = 0; i < coinList.length; i++) {  
            coin = money / coinList[i];  
            money = money % coinList[i];  
            result += String.format("%3d 원 동전의 수 : %2d 개\n", coinList[i], coin);  
        }  
        return result;  
    }  
}
```

Vending Machine

■ Change.JAVA

```
@Override
public String toString() {
    return "-----동전의 개수-----₩n" +
        "거스름돈 : " + String.format("%,d 원₩n", money) +
        change();
}
```

Vending Machine

■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) {  
        Scanner keyboard = new Scanner(System.in);  
        int price;    //물건의 가격  
        int pay;      //지불액  
  
        do {  
            System.out.print("물건의 가격 : ");  
            price = keyboard.nextInt();  
            System.out.print("지불액 : ");  
            pay = keyboard.nextInt();  
        } while (price <= 0 || pay <= 0 || pay - price < 0);    //do-while  
  
        Change change = new Change(pay - price);  
        System.out.println(change);  
    }  
}
```

체조 경기

- 올림픽의 체조 경기장에서 한 선수의 경기 내용을 7인의 채점관이 채점하여 최고 점수와 최소 점수를 삭제한 5명의 채점관 점수를 평균하여 점수로 한다.
 - 100.0점 만점으로 최소 점수는 60.0점 이상임



체조 경기(I)

■ Score 클래스

Score
-name : String -score[] : float
+input() : void +max() : float +min() : float +sum() : float +average() : float +info() : void

체조 경기(I)

■ 실행 화면

1번째	채점관	점수	입력	:	64.8
2번째	채점관	점수	입력	:	65.3
3번째	채점관	점수	입력	:	62.5
4번째	채점관	점수	입력	:	70.5
5번째	채점관	점수	입력	:	68.4
6번째	채점관	점수	입력	:	66.3
7번째	채점관	점수	입력	:	67.8
1번째	채점관	점수	:		64.80
2번째	채점관	점수	:		65.30
3번째	채점관	점수	:		62.50
4번째	채점관	점수	:		70.50
5번째	채점관	점수	:		68.40
6번째	채점관	점수	:		66.30
7번째	채점관	점수	:		67.80
최고 점수				:	70.5
최소 점수				:	62.5
이근식 선수 점수				:	66.52

체조 경기(I)

■ Score.JAVA

```
public class Score {  
    String name;  
    float[] score = new float[7];  
  
    public Score(String name) {  
        this.name = name;  
    }  
  
    private float max() {  
        float max = 60.0f;  
        for (int i = 0; i < score.length; i++) {  
            if (score[i] > max)  
                max = score[i];  
        }  
        return max;  
    }  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

체조 경기(I)

■ Score.JAVA

```
public void input() throws IOException {
    Scanner keyboard = new Scanner(System.in);
    int i = 0;
    while (i < score.length) {
        System.out.printf("%d번째 채점관 점수 입력 : ", i + 1);
        score[i] = keyboard.nextFloat();
        if (score[i] >= 60.0f && score[i] <= 100.0f)
            i++;
        else {
            System.err.println("오류");
            System.in.read();
        }
    }
}
```

체조 경기(I)

■ Score.JAVA

```
private float min() {  
    float min = 100.0f;  
    for (int i = 0; i < score.length; i++) {  
        if (score[i] < min)  
            min = score[i];  
    }  
    return min;  
}
```

```
private float sum() {  
    float total = 0.0f;  
    for (int i = 0; i < score.length; i++) {  
        total += score[i];  
    }  
    return total;  
}
```



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

체조 경기(I)

■ Score.JAVA

```
private float average() {  
    return (sum() - max() - min()) / 5;  
}
```

@Override

```
public String toString() {  
    String result = "";  
    for (int i = 0; i < score.length; i++)  
        result += String.format("%d번째 채점관 점수 : %5.2f\n", i + 1,  
                                score[i]);  
  
    result += String.format("최고 점수 : " + max());  
    result += String.format("최소 점수 : " + min());  
    result += String.format("%s 선수 점수 : %5.2f\n", name, average());  
  
    return result;  
}
```

체조 경기(I)

■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        Score lee = new Score("이근식");  
  
        lee.input();  
        System.out.println(lee);  
    }  
}
```

체조 경기(II)

■ Score.JAVA

```
public class Score {  
    private String name;  
    private Jumsu score;  
  
    public Score(String name) {  
        this.name = name;  
        score = new Jumsu();  
    }  
  
    private float max() {  
        float max = 60.0f;  
        for (int i = 0; i < score.getSize(); i++) {  
            if (score.getJumsu(i) > max)  
                max = score.getJumsu(i);  
        }  
        return max;  
    }  
}
```


체조 경기(II)

■ Score.JAVA

```
public void input() throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
  
    int i = 0;  
    while (i < score.getSize()) {  
        System.out.printf("%d번째 채점관 점수 입력 : ", i + 1);  
        score.setJumsu(i, keyboard.nextFloat());  
        if (score.getJumsu(i) >= 60.0f && score.getJumsu(i) <= 100.0f)  
            i++;  
        else {  
            System.err.println("오류");  
            System.in.read();  
        }  
    }  
}
```

Class 문제 12

■ Score.JAVA

```
private float min() {  
    float min = 100.0f;  
    for (int i = 0; i < score.getSize(); i++) {  
        if (score.getJumsu(i) < min)  
            min = score.getJumsu(i);  
    }  
    return min;  
}
```

```
private float sum() {  
    float total = 0.0f;  
    for (int i = 0; i < score.getSize(); i++) {  
        total += score.getJumsu(i);  
    }  
    return total;  
}
```

체조 경기(II)

■ Score.JAVA

```
private float average() {  
    return (sum() - max() - min()) / 5;  
}
```

@Override

```
public String toString() {  
    String result = "";  
    for (int i = 0; i < score.length; i++)  
        result += String.format("%d번째 채점관 점수 : %5.2f\n", i + 1,  
                                score[i]);  
  
    result += String.format("최고 점수 : " + max());  
    result += String.format("최소 점수 : " + min());  
    result += String.format("%s 선수 점수 : %5.2f\n", name, average());  
  
    return result;  
}
```

체조 경기(II)

■ Jumsu.JAVA

```
public class Jumsu {  
    private float[] jumsu = new float[7];  
  
    public void setJumsu(int i, float score) {  
        jumsu[i] = score;  
    }  
  
    public float getJumsu(int i) {  
        return jumsu[i];  
    }  
  
    public int getSize() {  
        return jumsu.length;  
    }  
}
```

체조 경기(II)

■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        Score lee = new Score("이근식");  
  
        lee.input();  
        System.out.println(lee);  
    }  
}
```

BMI

- BMI(Body Mass Index)는 카우프 지수로 비만을 평가하는 지수이며 세계적으로 쓰이는 공통 표준 지수임

남자 표준체중 = (키(Cm) - 100) * 0.9

여자 표준체중 = (키(Cm) - 100) * 0.85

BMI= 몸무게(Kg)/ 표준체중

내 체중 상태	계산한 체질량지수(BMI)
저체중	90 이하
정상체중	110이하
과체중	120이하
비만	경 도 130이하 중정도 150이하 고도 150초과

BMI

■ Data 클래스

```
public class Data {  
  
    static char dataRead(String prompt) throws IOException {  
        Scanner keyboard = new Scanner(System.in);  
        char data;  
        while (true) {  
            System.out.print(prompt);  
            data = keyboard.nextLine().charAt(0);  
            if (data == '남' || data == '여')  
                break;  
            else {  
                System.err.print("입력 데이터 오류");  
                System.in.read();  
            }  
        }  
        return data;  
    }  
}
```

BMI

■ Data 클래스

```
static double dataRead(String prompt, double index) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    double data;  
    while (true) {  
        System.out.print(prompt);  
        data = keyboard.nextDouble();  
        if (data > index)  
            break;  
        else {  
            System.err.print("입력 데이터 오류");  
            System.in.read();  
        }  
    }  
    return data;  
}
```


BMI

■ BMI 클래스

```
public class BMI {  
    private final char gender;  
    private final double weight;  
    private final double height;  
  
    public BMI(char gender, double weight, double height) {  
        this.gender = gender;  
        this.weight = weight;  
        this.height = height;  
    }  
  
    private double standard() {  
        double stdWeight;  
        if (gender == '남')  
            stdWeight = (height - 100) * 0.9;  
        else  
            stdWeight = (height - 100) * 0.85;  
        return stdWeight;  
    }  
}
```

BMI

■ BMI 클래스

```
private String check() {  
    int bmi = calcBMI();  
    String result = "BMI 지수 : " + bmi + "Wn";  
    if (bmi <= 90)  
        result += "저체중 입니다.";  
    else if (bmi <= 110)  
        result += "정상 체중 입니다.";  
    else if (bmi <= 120)  
        result += "과체중 입니다.";  
    else if (bmi <= 130)  
        result += "경도 비만 입니다.";  
    else if (bmi <= 150)  
        result += "중도 비만 입니다.";  
    else  
        result += "고도 비만 입니다.";  
    return result;  
}
```

BMI

■ BMI 클래스

```
private int calcBMI() {  
    return (int) (weight / standard() * 100);  
}
```

@Override

```
public String toString() {  
    return "신장 : " + height + "CmWn" +  
        "체중 : " + weight + "KgWn" +  
        check();  
}  
}
```

BMI

■ Main 클래스

```
public class Main extends Data {  
    public static void main(String[] args) throws IOException {  
        char gender;  
        double height;  
        double weight;  
  
        gender = dataRead("성별 입력 (남, 여) ");  
        height = dataRead("키를 입력 (단위, Cm) ", 120.0);  
        weight = dataRead("몸무게를 입력 (단위, Kg) ", 35.0);  
  
        BMI bmi = new BMI(gender, weight, height);  
        System.out.println(bmi);  
    }  
}
```

Bank Account

- 은행 계좌를 관리하는 프로그램을 만들어보자.



Bank Account

■ 은행 계좌 관리

Account
<ul style="list-style-type: none">-name : String (고객명)-no : String (계좌 번호)-balance : long (잔고)
<ul style="list-style-type: none">+getName() : String+getNo() : String+getBalance() : long+setName(new_name : String)+setNo(new_no : String)+setBalance(new_balance : long)+deposit(money : long)+withdraw(money : long)+toString()

Bank Account

■ Account.JAVA

```
public class Account {  
    private String name;    // 고객명  
    private String no;      // 계좌번호  
    private long balance;   // 잔액  
  
    public Account(String name, String no, long balance) {  
        this.name = name;  
        this.no = no;  
        this.balance = balance;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getNo() {  
        return (no);  
    }  
}
```

Bank Account

■ Account.JAVA

```
public long getBalance() {  
    return balance;  
}  
  
public void deposit(long money) { // 예금  
    balance += money;  
    System.out.println(name + " 계좌에 " +  
        String.format("%,d원 입금했습니다", money));  
}  
  
public void withdraw(long money) { // 출금  
    if (balance < money)  
        System.out.println(name + " 계좌 잔고가 부족하여 출금이 어렵습니다");  
    else {  
        balance -= money;  
        System.out.println(name + " 계좌에 " +  
            String.format("%,d원 출금했습니다", money));  
    }  
}
```


Bank Account

■ Account.JAVA

@Override

```
public String toString() {  
    return "계좌주 : " + name +  
        ", 계좌 번호 : " + no +  
        ", 잔액 : " + String.format("%,d 원", balance);  
}  
}
```

Bank Account

■ Main.JAVA

```
public static void main(String[] args) {  
    Account[] account = new Account[3];  
    /*    account[0] = new Account("박철수", "123456", 1000);  
        account[1] = new Account("이영희", "654321", 200);  
        account[2] = new Account("홍길동", "111111", 0); */  
  
    account = new Account[] {new Account("박철수", "123456", 1000),  
                             new Account("이영희", "654321", 200),  
                             new Account("홍길동", "111111", 0)};  
  
    System.out.println("---계좌정보---");  
    for (int i = 0; i < account.length; i++){  
        System.out.println(account[i]);  
    }  
}
```

Bank Account

■ Main.JAVA

```
account[0].withdraw(200);
account[1].deposit(100);
account[2].deposit(account[0].getBalance() + account[1].getBalance());
account[0].withdraw(account[0].getBalance() + 100);
account[1].withdraw(account[1].getBalance());

System.out.println("---계좌정보---");
for (int i = 0; i < account.length; i++){
    System.out.println(account[i]);
}
long totalBalance = 0L;
for (int i = 0; i < account.length; i++){
    totalBalance += account[i].getBalance();
}
System.out.println("우리은행의 모든 계좌의 총 잔액은 " +
    String.format("%,d원", totalBalance) + " 입니다.");
}
```

Bank Account2

■ Account.JAVA

```
public class Account {  
    private String name; // 고객명  
    private String no;   // 계좌번호  
    private long balance; // 잔액  
  
    public Account(String name, String no, long balance) {  
        this.name = name;  
        this.no = no;  
        this.balance = balance;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getNo() {  
        return no;  
    }  
}
```

Bank Account2

■ Account.JAVA

```
public long getBalance() {  
    return balance;  
}  
  
public void deposit(long money) { // 예금  
    balance += money;  
}  
  
public boolean withdraw(long money) { // 출금  
    if (balance < money)  
        return false;  
    else {  
        balance -= money;  
        return true;  
    }  
}
```

Bank Account2

■ Account.JAVA

@Override

```
public String toString() {  
    return "계좌주 : " + name +  
        ", 계좌 번호 : " + no +  
        ", 잔액 : " + String.format("%,d 원", balance);  
}  
}
```

Bank Account2

■ Main.JAVA

```
public class Main {  
    private static Account[] account = new Account[100];  
    private static Scanner keyboard = new Scanner(System.in);  
    public static int cnt = 0;  
  
    public static void main(String[] args) {  
        while (true) {  
            System.out.println("-----");  
            System.out.println("1. 계좌생성 | 2. 계좌목록 | 3. 예금 | 4. 출금 | 5. 종료");  
            System.out.println("-----");  
            System.out.print("선택 > ");  
            int select = keyboard.nextInt();  
            if (select == 1) {  
                createAccount();  
            } else if (select == 2) {  
                accountList();  
            } else if (select == 3) {  
                deposit();  
            }  
        }  
    }  
}
```

Bank Account2

■ Main.JAVA

```
        } else if (select == 4) {  
            withdraw();  
        } else if (select == 5) {  
            break;  
        }  
    }  
    System.out.println("프로그램 종료");  
}  
private static void menu(char mark, String title) {  
    line(mark, title.length() + 10);  
    System.out.println(title);  
    line(mark, title.length() + 10);  
}  
private static void line(char mark, int count) {  
    for (int i = 0; i < count; i++)  
        System.out.print(mark);  
    System.out.println();  
}
```


Bank Account2

■ Main.JAVA

```
private static void createAccount() {  
    menu('*', "계좌 생성");  
    System.out.print("계좌 번호 : ");  
    String no = keyboard.next();  
    System.out.print("계좌주 : ");  
    String name = keyboard.next();  
    System.out.print("초기입금액: ");  
    int balance = keyboard.nextInt();  
  
    account[cnt] = new Account(name, no, balance);  
    cnt++;  
    System.out.println("계좌가 생성되었습니다.");  
}
```

Bank Account2

■ Main.JAVA

```
private static void accountList() {  
    menu('#', "계좌목록");  
    if (cnt == 0)  
        System.out.println("계좌가 없습니다");  
    else{  
        for (int i = 0; i < cnt; i++) {  
            System.out.println(account[i]);  
        }  
    }  
}
```

Bank Account2

■ Main.JAVA

```
private static void deposit() {  
    menu('-', "예 금");  
    System.out.print("계좌번호: ");  
    String no = keyboard.next();  
    System.out.print("예금액: ");  
    int money = keyboard.nextInt();  
    for (int i = 0; i < cnt; i++) {  
        System.out.println(no);  
        System.out.println(account[i].getNo());  
        if (no.equals(account[i].getNo())) {  
            account[i].deposit(money);  
            System.out.println(account[i].getName() + " 계좌에 " +  
                String.format("%,d원 입금하여 잔고가 %,d 원 입니다",  
                    money, account[i].getBalance()));  
        } else {  
            System.out.println("입력하신 계좌번호가 존재하지 않습니다.");  
        }  
    }  
}
```

Bank Account2

■ Main.JAVA

```
private static void withdraw() {
    menu('-', "출 금 내 용 ");
    System.out.print("계좌번호: ");
    String no = keyboard.next();
    System.out.print("출금액: ");
    int money = keyboard.nextInt();
    for (int i = 0; i < cnt; i++) {
        if (no.equals(account[i].getNo())) {
            if (account[i].withdraw(money))
                System.out.println(account[i].getName() + " 계좌에 " +
                    String.format("%,d원 출금하여 잔고가 %,d 원 입니다",
                        money, account[i].getBalance()));
            else
                System.out.println("잔고 부족으로 출금이 거부되었습니다.");
        } else
            System.out.println("입력하신 계좌번호가 존재하지 않습니다.");
    }
}
```