

프로그래밍 기초

JAVA

변수 타입

정수 : byte(1b), short(2b), int(4b), long(8b)

실수 : float(4b), double(8b)

문자 : char(2b)

부울 : boolean (1b)

정수리터럴

8진수 : 016, 014

16진수 : 0xe, 0x10, 0x11

10진수 : 10, 4, 12

10진수 : 5l , 24L,3l

소수리터럴

double: 0.34, 12.3

float : 3.14f, 0.14f

문자형

'가' , 'a' , '\uac00'

문자열내 특수문자

\w, \b, \r ,\n, \', \", \uac00

키워드

abstract – 추상

new – new heap space

public – 접근자

default – 접근자

private – 접근자

protected – 접근자

this – 자신객체

implements – 구현

extends – 상속,확장

throw – 예외발생

try ~ catch ~finally 예외처리

final – 상속불가 클래스

enum – 문자숫자매칭

instanceof - ~의 인스턴스

synchronized – 동기화



관례적 코딩기법

클래스 이름의 첫글자는 항상 대문자로 시작한다.
메소드, 변수등의 이름은 첫글자를 항상 소문자로 시작한다.
상수는 모든문자를 대문자로 한다.연결은 _
카멜기법을 사용한다.

연산자 C 언어와 동일 :
쉬프트연산자 >>> 부호비트까지 0으로 채움

클래스 class

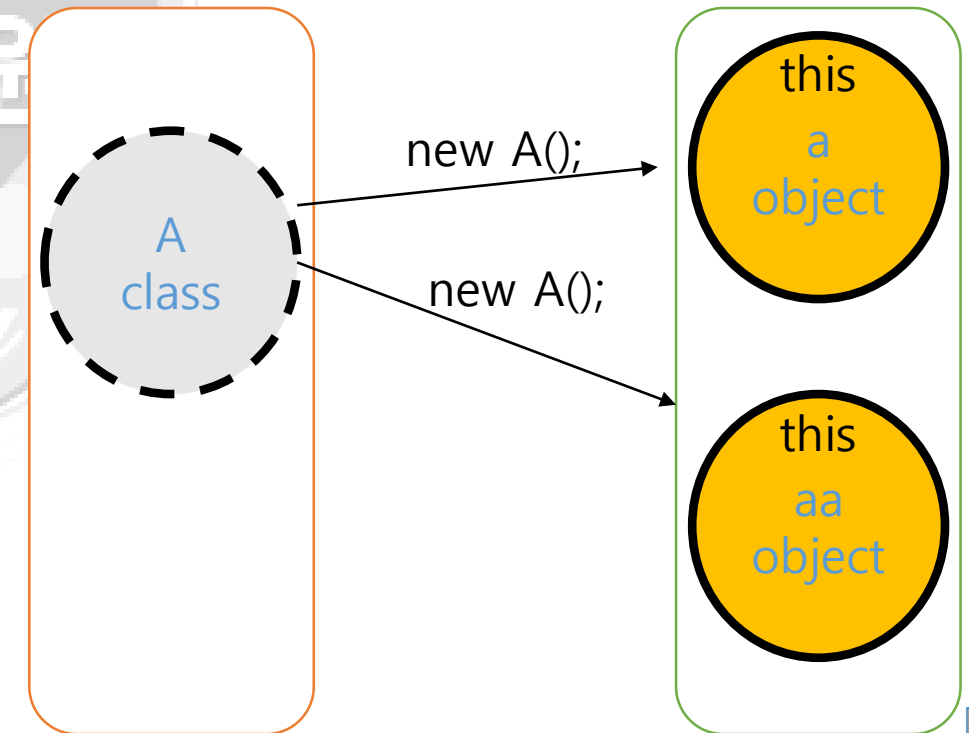
객체를 만들어내는 도구, 객체를 만들어 내는 공장

객체 Object

클래스가 메모리에 인스턴스화 되어 만들어 내는 생산품

static
메모리영역

heap
메모리영역



클래스 생성

클래스는 항상 스택 영역에 존재하며 실행시 객체를 만들어 내며 객체는 항상 힙영역에 존재합니다.
this 란 키워드는 항상 객체 자신을 가르키며 클래스를 가르키지 않습니다.
객체는 자신을 생성한 클래스를 알고 있습니다.

```
class A {  
    int a=0;  
    int method(){  
        ...  
    }  
}
```

클래스명

멤버변수, 객체변수

메소드

a 변수명과 method 명은 this 키워드가 생략된것으로 보아도 무방함
객체의 변수 a , 객체의 메소드 method
a 변수와 method 는 객체가 생성되어야 메모리 공간이 확보되어 사용할 수 있음

객체의 생성

객체의 생성 시점을 잘 알아야 됩니다.

클래스는 프로그램 로딩시 자동으로 스택 영역에 탑재되며

new 키워드를 사용하여 객체를 생성하였을때 heap 영역에 생성되어 메모리 공간을 갖게 됩니다.

객체가 생성되지 않은 상태에서 메모리 공간에 접근(즉, 변수에 접근) 하게 되면 오류가 발생합니다.

```
A abc = new A(); print.out(abc.a);  
A abb = new A(); abb.method();
```

참조형, 레퍼런스

new 키워드 사용시 힙영역에 객체를 만들어 내고 객체를 가르키는 레퍼런스(포인터)를 반환합니다.
크기를 가늠할 수 없는 배열, 객체는 모두 레퍼런스 방식을 사용합니다.

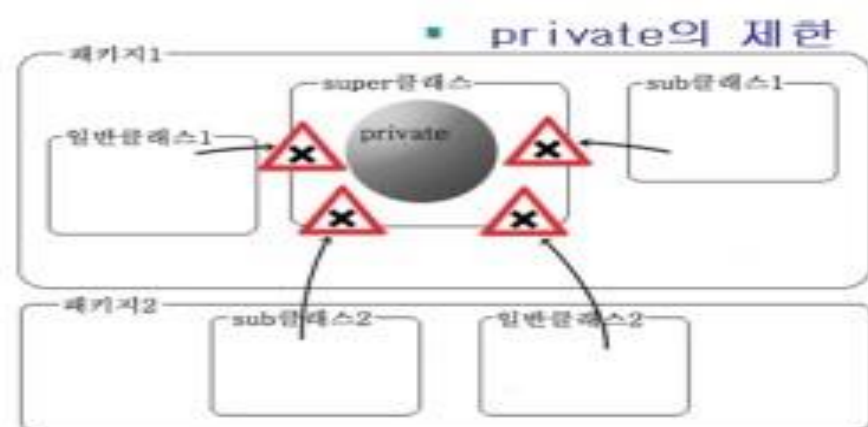
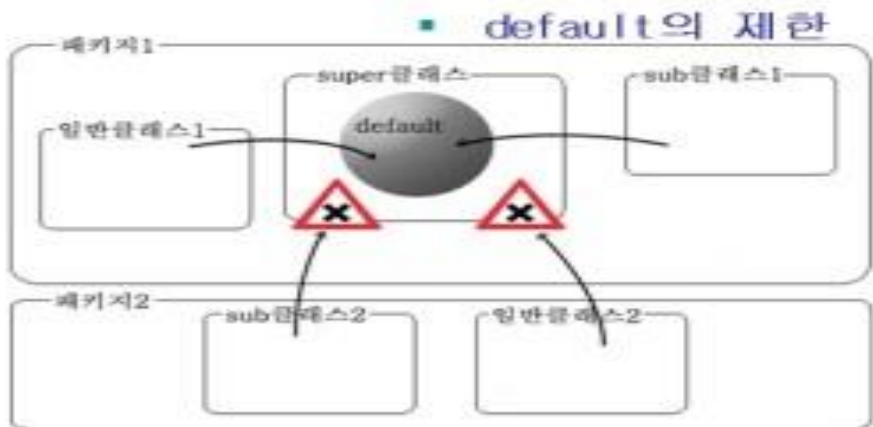
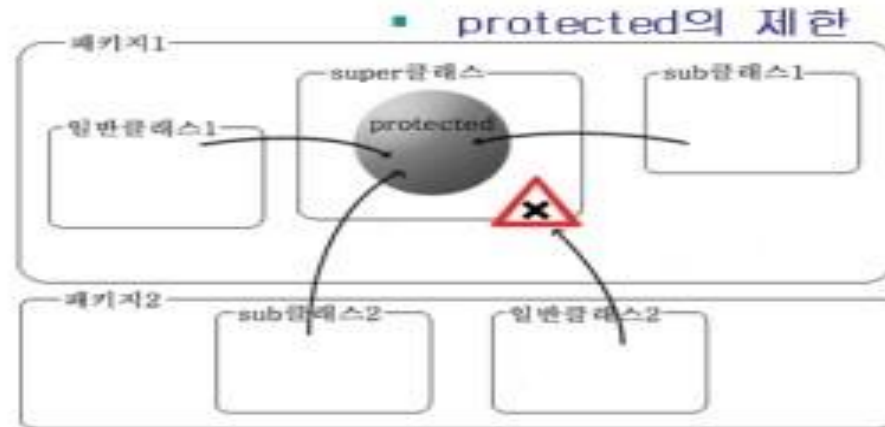
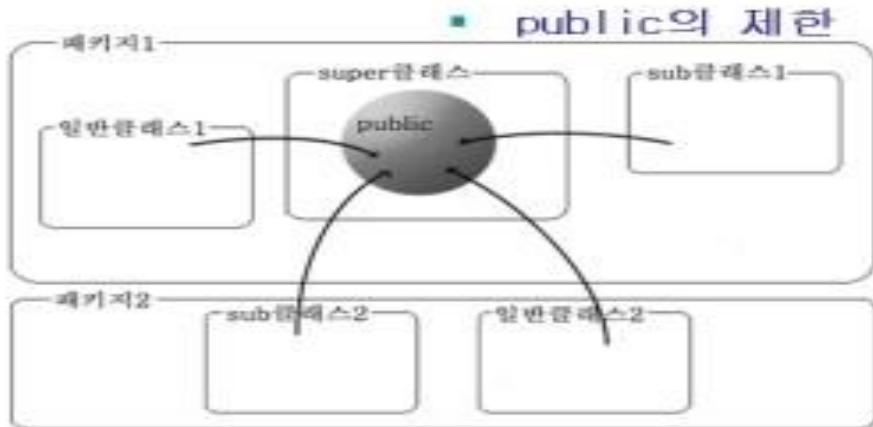
접근자 클래스, 멤버변수 및 메소드 접두어에 사용되며 기본값은 default 입니다.

public – 동일 클래스, 외부 클래스, 동일패키지, 외부패키지등 모든곳에서 접근할 수 있습니다.

private – 동일 클래스 내에서만 접근 가능합니다. 외부접근 차단

protected – 동일 패키지과 상속관계에서 접근 가능합니다.

default – 동일 패키지에서만 접근 가능합니다.



접근제한자 사용

- 클래스 : public, default
- 생성자 : public, protected, default, private
- 멤버변수 : public, protected, default, private
- 멤버메소드 : public, protected, default, private
- 지역변수 : 접근제한자 사용 불허

getter 와 setter

```
class Variable{  
    private int a;  
  
    public void setA(int n){  
        a=n;  
    }  
  
    public int getA(){  
        return a;  
    }  
}
```

데이터은닉화:외부노출안됨

캡슐화 : 메소드에의해 값 셋팅 및 리턴

클래스의 스택 영역

객체 생성전 영역이므로 객체에서는 활용할 수 있으나 스택영역에서 객체를 활용하면 안된다.

```
class A{
```

```
    static int a=0;
```

```
    int b = 1;
```

```
    static int method1(){
```

```
        System.out.print (a);
```

```
        System.out.print (b);
```

```
    }
```

```
}
```

a 변수영역에 접근가능

static
메모리 영역

heap
메모리 영역

b 변수영역에 접근오류

new A();

A
class
int a
method1

a
object
int b

static 메소드나 변수는 객체를 만들어 내지 않고
클래스명으로 접근하여 사용가능하다.

배열

```
int[] arr = new int[5];  
arr = {1,2,3,4,5};  
arr[0] = 3;  
System.out.print(arr[0]);  
int[] arrRef = arr;
```

선언과 동시에 초기화

```
int[] score = {10, 20, 30, 40, 50}
```

선언후 초기화

```
int[] score;  
score = new int[]{10, 20, 30, 40, 50}
```

2차원 배열

```
int[][] arr = {{3,4},{5,6,7},{8,9}}  
int[][] arr1 = new int[3][];  
arr1[0] = new int[2];  
arr1[1] = new int[3];
```

문자배열

```
String[] strArr = {"일","이","삼"};
```

클래스 배열

```
class A{
```

```
A[] arrA = new A[3];
```

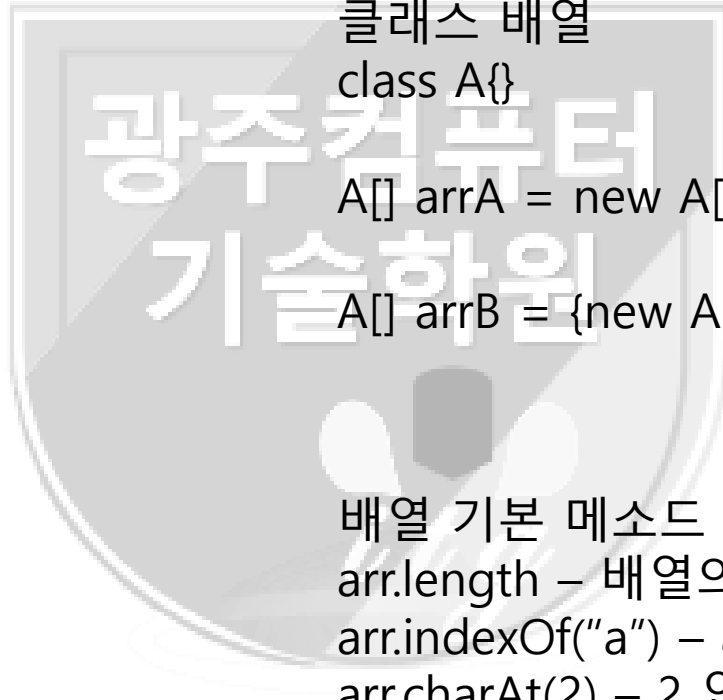
```
A[] arrB = {new A(), new A(), new A()}
```

배열 기본 메소드

arr.length – 배열의 길이 출력

arr.indexOf("a") – a 문자의 인덱스 출력

arr.charAt(2) – 2 인덱스 캐릭터 출력



참조형 변수와 타입

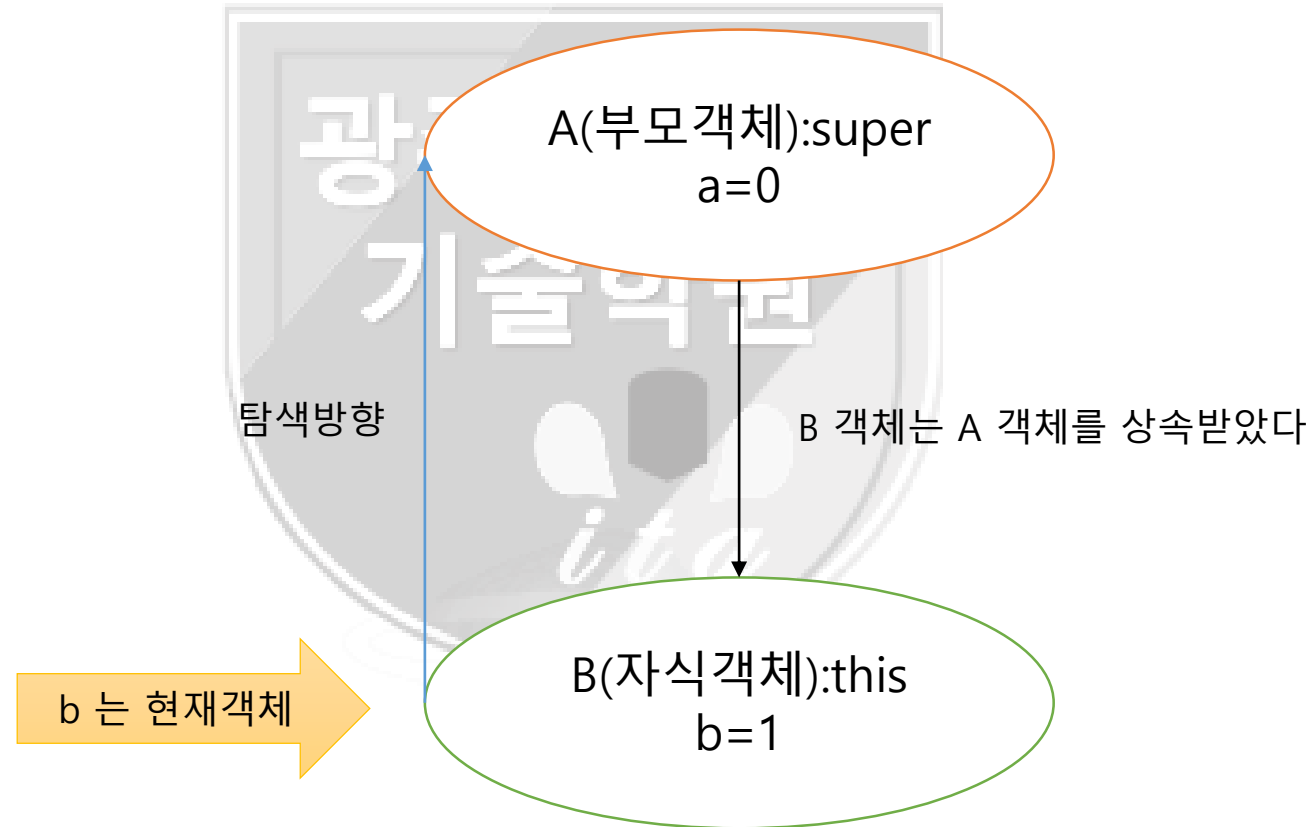
클래스는 타입이다

String 클래스 – 문자를 배열형태로 담는 문자배열, 문자열 클래스

상속 : 상속받은 자식객체 생성시 부모객체도 같이 생성된다.

```
class A{  
    int a;  
    int getA(){  
        return a;  
    }  
}  
class B extends A{  
    int b;  
}
```

B b = new B();



System.out.print(b.a) :: 0 출력

System.out.print(b.getA()) :: 0 부모메소드 실행

자기자신의 멤버변수/메소드를 먼저 검색후 없다면 부모 멤버변수/메소드를 검사한다.

생성자

```
class A{  
    int a;  
    public A(int a){this.a = a;}  
}
```

```
A a = new A(10);  
a.a :: 10
```

오버로딩

```
class A{  
    int a;int b;  
    public A(int a){this.a = a;}  
    public A(int a, int b){  
        this.a=a;this.b=b;  
    }  
}
```

```
A a = new A(10,20);  
a.b :: 20  
a.a :: 10
```

오버라이드

```
class A{  
    int a;  
    int am(int b){  
        System.out.print("부모메소드")  
    }  
}
```

```
class B extends A{  
    int a;  
    int am(int b){  
        System.out.print("자식메소드")  
    }  
}
```

new B().am(3) :: 자식메소드

참조형 파라미터

```
class A{  
    int test(B b){  
        b.bb = 500;  
        return b.bb;  
    }  
    B test1(B b){return b}  
}
```

```
class B{  
    int bb = 100;  
}
```

new A().test(new B()) :: 500

new A().test1(new B()).bb :: 100

추상화

```
abstract class A{  
    int a;  
    int b=3;  
    void setA(int a){  
        this.a = a;  
    }  
    abstract void setB(int b);  
}
```

추상메소드

구현되지 않은 추상메소드를 가지고 있으면 추상클래스로 선언해야한다.

```
class B extends A{  
    int b=5;  
    void setB(int b){  
        this.b = b;  
        super.b = b;  
    }  
}
```

```
B b = new B();  
b.setB(10);  
b.setA(20);
```

추상클래스를 상속받은 자식클래스는 추상클래스가 아니라면 추상메소드를 반드시 구현해야한다.

인터페이스

인터페이스는 설계이며 다형성의 타입이다.

객체를 만들지 않으므로 객체변수를 쓸수 없고 메소드는 모두 추상메소드 이다

```
interface Afun{  
    static final String funstr = "TV전원"  
    void onOffTv(int val);  
}
```

```
interface Bfun{  
    static final String funstr = "채널전환"  
    void changeChannel(int chanNum);  
}
```

```
interface Cfun{  
    static final String funstr = "볼륨조정"  
    void controlVol(int vol);  
}
```

JDK 8 이상부터는 인터페이스에 정적 메소드를 작성할 수 있다.

```
class ABC implements Afun, Bfun, Cfun{  
    void onOffTv(int val){...}  
    void changeChannel(int chanNum){...}  
    void controlVol(int vol){...}  
}
```

인터페이스는 다중 구현이 가능하며 인터페이스끼리의 상속도 가능하다
구현시 추상메소드를 모두 구현해야하며 일부 구현이 안되면 추상클래스로 설정해야한다.

```
Afun abc = new ABC();  
Bfun abb = new ABC();  
Cfun abb = new ABC();  
ABC aaa = new ABC();
```

인터페이스의 다형성

무명클래스-class를 1회성으로 구현

```
Remote r = new Remote(){int a=0;void meth(){}};
```

내부클래스

```
class A{  
    class B{ }  
    B retB(){  
        return new B();  
    }  
}
```

```
B b = new A().retB();
```

예외처리

```
try{ 정상처리...}catch(Exception e){ 예외발생...} finally{예외상관없이 실행...}
```



Thread

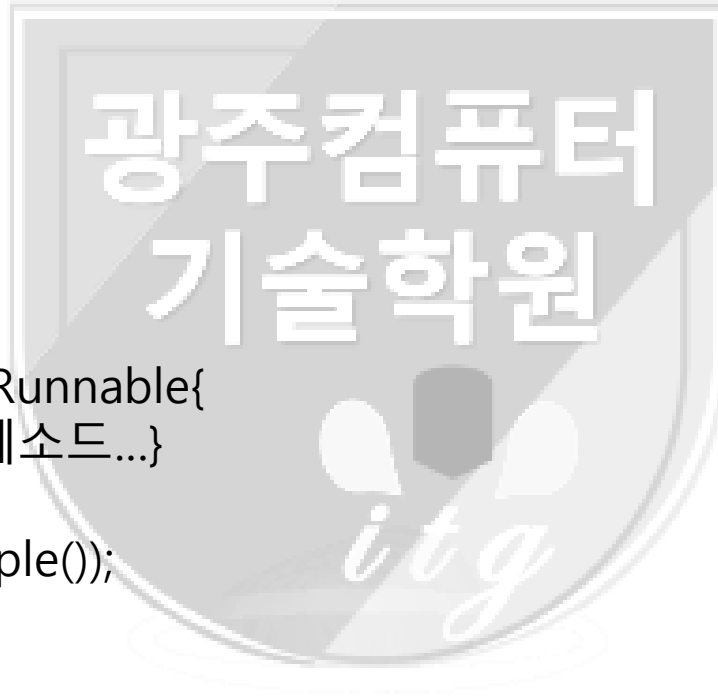
```
public class Sample extends Thread{  
    public void run(){... 쓰레드코드...}  
}
```

```
Thread t = new Sample();  
t.start()
```

Runnable

```
public class Sample implements Runnable{  
    public void Run(){...오버라이드메소드...}  
}
```

```
Thread t = new Thread(new Sample());
```



JAVA jdbc 연결

```
import java.sql.*;
public class WDB {
public static void main(String[] args){
    //드라이버 로드
    Connection con = null;
    ResultSet rs = null;
    PreparedStatement psmt = null;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        String url = "jdbc:oracle:thin:@localhost:1521:orcl";
        String id = "c##web";
        String pw = "oracle";
        System.out.println("DB 정상연결");
        try {
            con = DriverManager.getConnection(url,id,pw);
            System.out.println("DB 계정일치");
        } catch (SQLException e) {
            System.out.println("DB 계정불일치");
            e.printStackTrace();
        }
    } catch (ClassNotFoundException e) {
        System.out.println("DB 연결실패");
        e.printStackTrace();
    }
}
```

```
try {
    String sql = "insert into department(deptno, deptname,floor) values (?,?,?)";
    int deptno = 1;
    String deptname = "deptname";
    int floor = 1;
    pstmt = con.prepareStatement(sql);
    pstmt.setInt(1, deptno);
    pstmt.setString(2, deptname);
    pstmt.setInt(3, floor);
    pstmt.executeUpdate();
    System.out.println("쿼리성공");
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("쿼리실패");
}
```



```
try {
    int floor =1;
    String sql = "select * from department where floor=?";
    psmt = con.prepareStatement(sql);
    psmt.setInt(1, floor);
    rs = psmt.executeQuery();
    while(rs.next()) {
        int a = rs.getInt("deptno");
        String b = rs.getString("deptname");
        int c = rs.getInt("floor");
        System.out.println("deptno은 "+a+"deptname은 "+b+"floor은 "+c);
    }
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("쿼리실패");
}
```

13. 다음은 자바 소스코드이다. 출력 결과를 쓰시오.

```
public class good {  
    public static void main(String[] args){  
        int i;  
        int a[4] = {0,1,2,3};  
        for(i=0; i<4; i++){ system.out.print(a[i] + ""); }  
    }  
}
```

14. 다음은 자바 소스코드이다. 출력 결과를 쓰시오.

답기
-8

```
public class good {  
    public static void main(String[] args){  
        int i = 3;  
        int k = 1;  
        swich (i) {  
            case 0; case 1; case 2;  
            case 3 : k = 0;  
            case 4 : k += 3;  
            case 5 : k -= 10;  
            default: k--;  
        }  
        system.out.print(k); }  
}
```



5. 다음은 자바 코드이다. 다음 밑줄에 들어갈 키워드를 쓰시오.

달기
new

```
class parent
{
    public void show(){
        system.out.println("Parent");
    }
}
class Child extends Parent{
    public void show(){
        system.out.println("Child");
    }
}
public class good{
    public static void main(String[] args){
        Parent pa = ____ Child();
        pa.show();
    }
}
```



19. 다음은 자바 코드이다. 출력 결과를 쓰시오.

닫기
a = 10

```
class A{
    private int a;
    public A(int a){
        this.a = a;
    }
    public void display(){
        system.out.println("a=" + a);
    }
}

class B extends A {
    public B(int a){
        super(a);
        super.display();
    }
}

public class good {
    public static void main(String[] args){ B obj = new B(10); }
}
```



15. 다음은 자바 코드이다. 출력 결과를 쓰시오.

```
abstract class vehicle{
    private String name;
    abstract public String getName(String val);
    public String getName(){
        return "vehicle name:" + name;
    }
    public void setName(String val){
        name = val;
    }
}

class Car extends Vehicle{
    public Car(String val){
        setName(val);
    }
    public String getName(String val){
        return "Car name : " + val;
    }
    public String getName(byte val[]){
        return "Car name : " + val;
    }
}

public class good {
    public Static void main(String[] args){
        Vehicle obj = new Car("Spark");
        System.out.print(obj.getName()); } }
```

달기

Vehicle name : Spark



17. 다음은 자바 코드이다. 출력 결과를 쓰시오.

```
public class good {  
    public static void main(String[] args){  
        int i=0;  
        int sum=0;  
        while (i<10){  
            i++;  
            if(i%2 ==1) continue;  
            sum += i;  
        }  
        System.out.println(sum); }  
}
```

2+4+6+8+10=30이 됩니다.

닫기
30

0/2 = 0
2/0 오류



5. 다음은 n이 10일 때, 10을 2진수로 변환하는 자바 소스 코드이다. 1,2에 알맞는 값을 적으시오.

[출력결과] : 00001010

닫기

1. $n > 0$ or $n \geq 1$ or $i < 8$ or $i \leq 7$

2. $n \% 2$ or $n \& 1$

```
class good {  
    public static void main (String[] args) {  
        int[] a = new int[8];  
        int i=0;  
        int n=10;  
        while ( 1. ) {  
            a[i++] = ( 2. );  
            n /= 2;  
        }  
        for(i=7; i>=0; i--){  
            System.out.print(a[i]);  
        }  
    }  
}
```



6. 다음은 자바 소스 코드이다. 출력 결과를 보고 , 1,2에
알맞는 값을 적으시오.

[출력 결과]

1 4 7 10 13

2 5 8 11 14

3 6 9 12 15

```
public class good {  
    public static void main(String[] args) {  
        int[][]a = new int[(1.)][(2.)];  
        for(int i = 0; i <3; i++){  
            for(int j=0; j < 5; j++){  
                a[i][j] = j*3+(i+1);  
                System.out.print(a[i][j]+"");  
            }  
            System.out.println();  
        }  
    }  
}
```

닫기

1. 3

2. 5

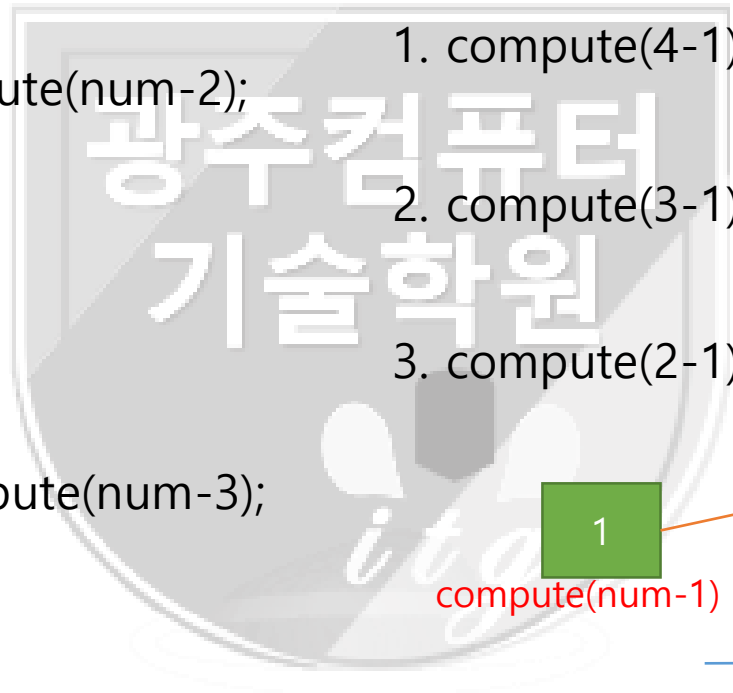


19. 다음은 자바 소스 코드이다. 출력 결과를 쓰시오.

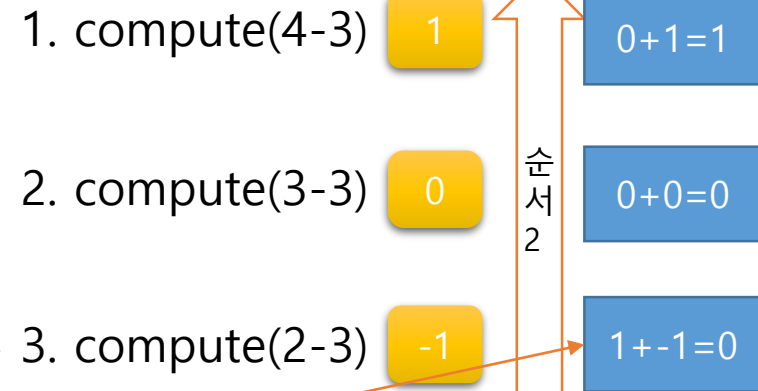
```
class Parent{
    public int compute(int num){
        if(num <=1) return num;
        return compute(num-1) + compute(num-2);
    }
}
```

```
class Child extends parent {
    public int compute(int num){
        if(num<=1) return num;
        return compute(num-1) + compute(num-3);
    }
}
```

```
class Exam{
    public static void main (String[] args){
        Parent obj = new Child();
        System.out.print(obj.compute(4)); }
}
```



stack 저장부분



compute(num-1) 결과에
compute(num-3) 결과를
더하기 수행

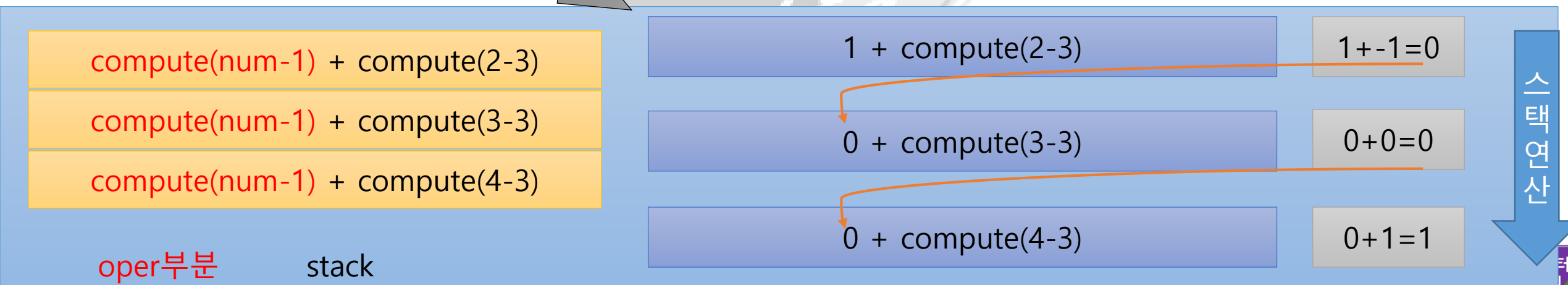
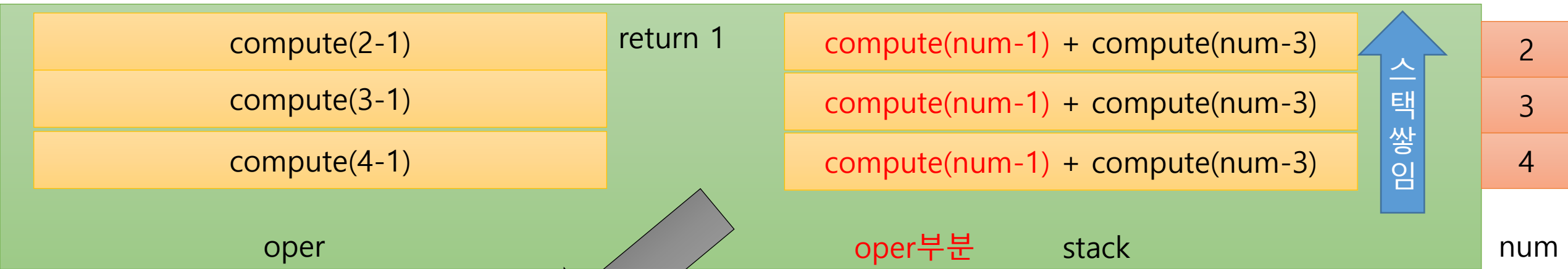
연산후 +

```

class Child extends parent {
    public int compute(int num){
        if(num<=1) return num;
        return compute(num-1) + compute(num-3);
    }
}

```

num 이 스택에 쌓일때 현재 num 적용됨



7. 다음 Java 프로그램 결과를 쓰시오.

```
public class good{  
    public static void main(String[] args){  
        int[][]arr = new int[][]{{45,50,75},{89}};  
        System.out.println(arr[0].length);  
        System.out.println(arr[1].length);  
        System.out.println(arr[0][0]);  
        System.out.println(arr[0][1]);  
        System.out.println(arr[1][0]);  
    }  
}
```

닫기

3

1

45

50

89



17. 다음은 Java 프로그램이다. 실행 결과를 쓰시오.

```
public class good {  
    public static void main(String[] args){  
        int i, j;  
        for(j=0, i=0; i<=5; i++){  
            j+=i;  
            System.out.print(i);  
            if(i==5){  
                System.out.print("=");  
                System.out.print(j);  
            }else{  
                System.out.print(" ");  
            }  
        }  
    }  
}
```

답기

0 + 1 + 2 + 3 + 4 + 5 = 15

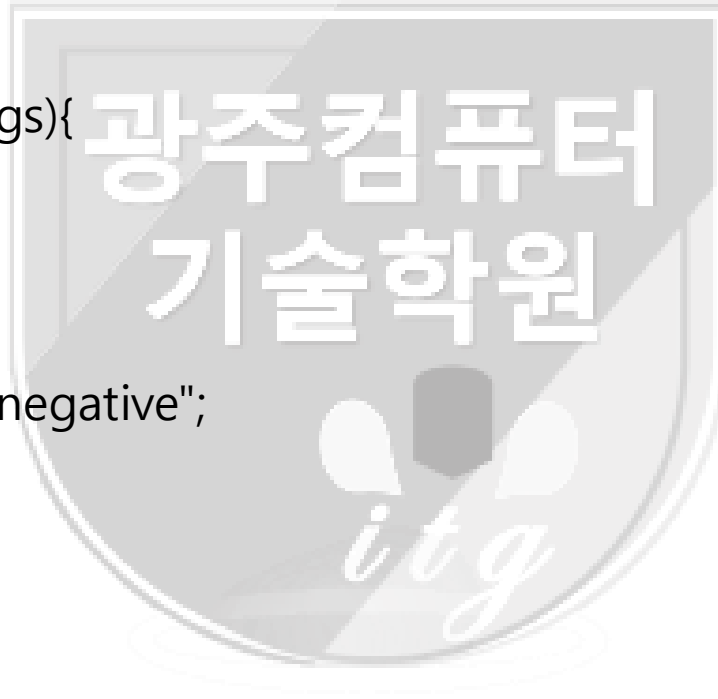


17. 클래스 내에서 객체 생성 없이 사용할 수 있는
메소드로써 출력 결과를 작성하시오.

닫기
static

```
public class Test {  
    public static void main(String[] args){  
        system.out.print(test.check(1));  
    }  
  
    ( ) String check (int num) {  
        return (num >= 0) ? "positive" : "negative";  
    }  
}
```

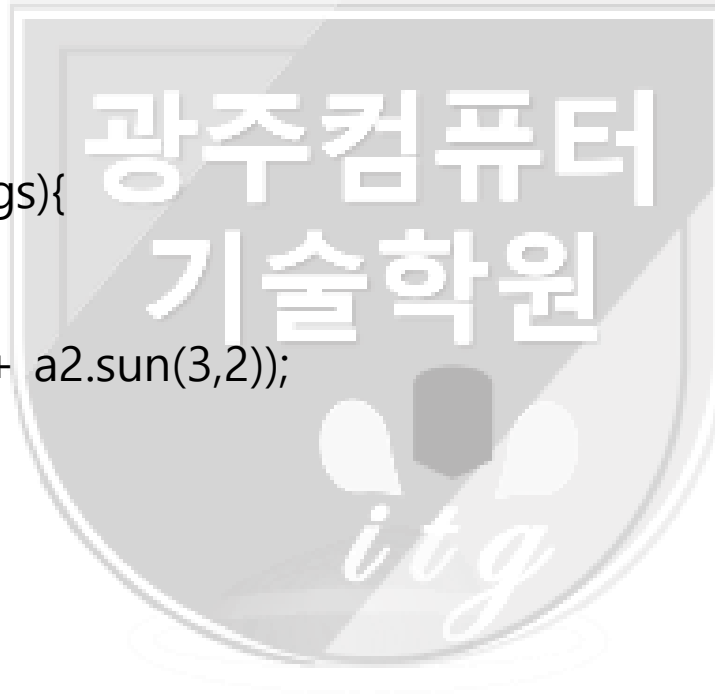
[출력결과] positive



19. 다음은 JAVA 관한 문제이다. 알맞는 출력값을 작성하시오.

답기
11

```
public class over1 {  
    public static void main(String[] args){  
        ovr a1 = new ovr1();  
        ovr a2 = new ovr2();  
        System.out.println(a1.sun(3,2) + a2.sun(3,2));  
    }  
    int sun(int x, int y){  
        return x + y;  
    }  
}  
class ovr2 extends ovr1 {  
    int sun(int x, int y){  
        return x - y + super.sun(x,y);  
    }  
}
```



11. 다음 Java 코드에 대한 알맞는 출력값을 쓰시오.

```

public class testco {
    public static void main(String[] args) {
        int a = 3, b = 4, c = 3, d = 5;
        if((a == 2 | a == c) & !(c > d) & (1 == b ^ c != d)) {
            a = b + c;
            if(7 == b ^ c != a) {
                System.out.println(a);
            } else {
                System.out.println(b);
            }
        } else {
            a = c + d;
            if(7 == c ^ d != a) {
                System.out.println(a);
            } else {
                System.out.println(d);
            }
        }
    }
}

```

닫기
7

$\text{true} \ \& \ \text{true} \ \& \ (\text{false} \wedge \text{true})$
 $\text{true} \quad \& \quad \text{true}$
 res true

$\text{false} \wedge \text{true}$
 res true

2022년 1회

3. 다음 자바 문법에 알맞는 출력 결과를 작성하시오.

```
class A {  
    int a;  
    int b;  
}  
  
public class Main {  
    static void func1(A m){  
        m.a *= 10;}  
    static void func2(A m){  
        m.a += m.b; }  
    public static void main(String args[]){  
        A m = new A();  
        m.a = 100;  
        func1(m);  
        m.b = m.a;  
        func2(m);  
        System.out.printf("%d", m.a); }  
}
```

달기
2000



11. 다음 Java 코드 중에서 밑줄에 들어갈 알맞는 코드를 작성하시오.

```
class Car implements Runnable{  
    int a;  
    public void run(){  
        system.out.println("message")  
    }  
}  
  
public class Main{  
    public static void main(String args[]){  
        Thread t1 = new Thread(new ____());  
        t1.start();  
    }  
}
```



7. 다음 자바 코드에 알맞는 출력값을 작성하시오.

-8

```
public static void main(String args[]){  
    int i = 3; int k = 1;  
    switch(i) {  
        case 1: k += 1;  
        case 2: k++;  
        case 3: k = 0;  
        case 4: k += 3;  
        case 5: k -= 10;  
        default: k--;  
    }  
    System.out.print(k);  
}
```



17. 다음 자바코드에서 알맞는 출력값을 작성하시오.

```
public class Conv{
    public Conv(int a) {
        this.a = a;
    }
    int func() {
        int b = 1;
        for (int i=1; i<a; i++){
            b = a * i + b
        }
        return a +b;
    }
    int a;
}

public static void main(String[] args){
    Conv obj = new Conv(3);
    obj.a=5;
    int b = obj.func();
    system.out.print(obj.a + b);
    i
}
```

답기
61



6+16+31+51