

# Lesson 4 – Object, Class , Instance, Method , Constructor -1

2005 2

# @ Lesson 4 – Object, Class, Instance, Method, Constructor - 1



**(Object)**

**(Class)**

**(Instance)**

**(Method)**

**(Constructor)**

**Memory      1**

**UML 1**

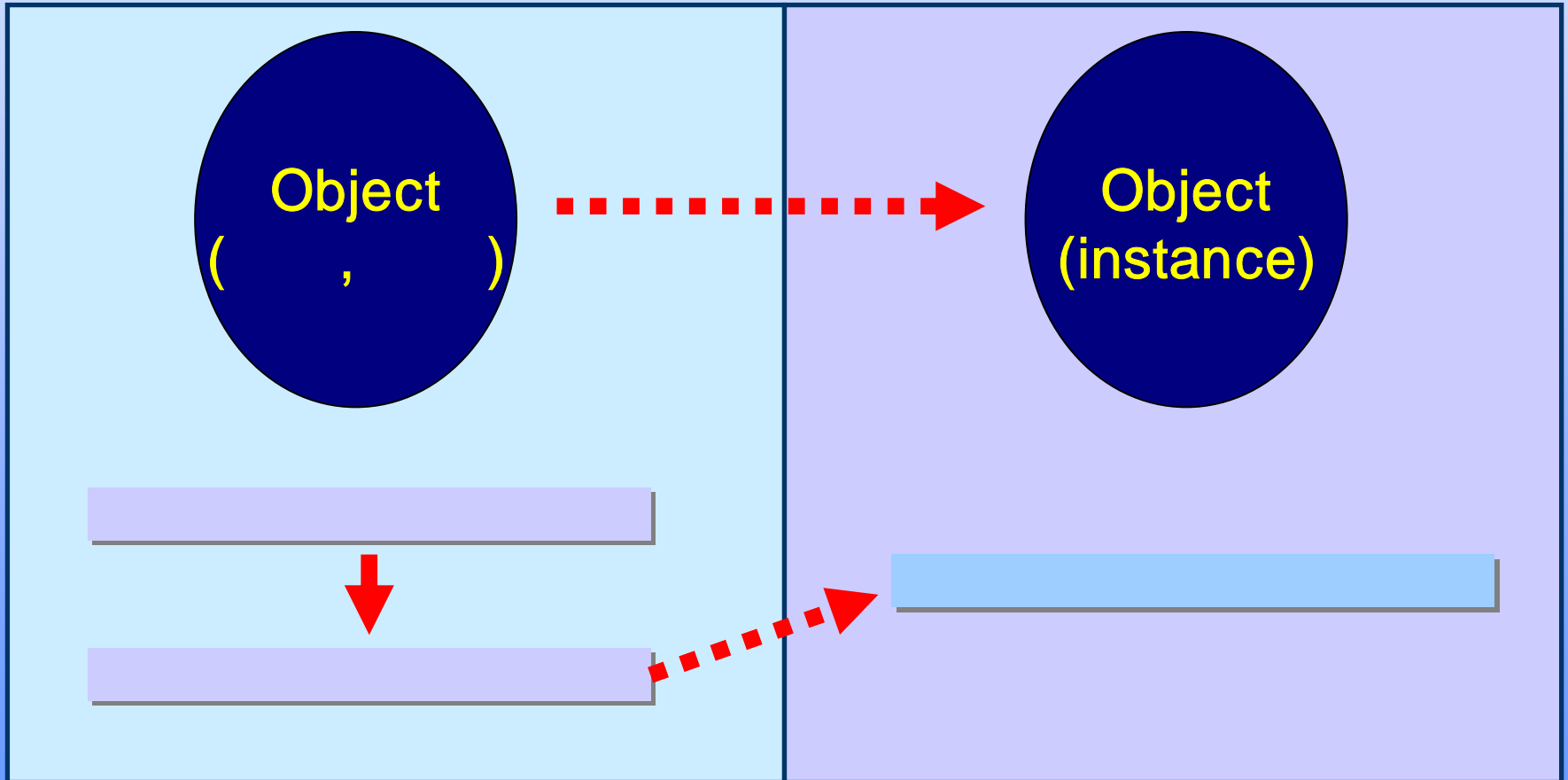
@

&

1

(Real World)

(Software World)



@

&

가

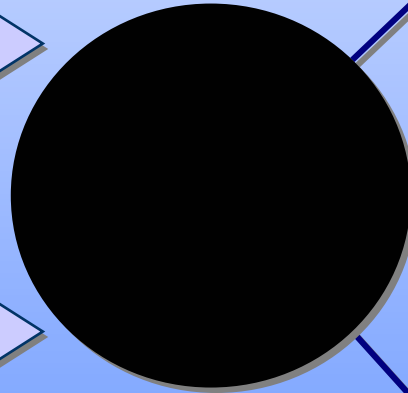
.

.

가

( )

( )



가

@

&

2

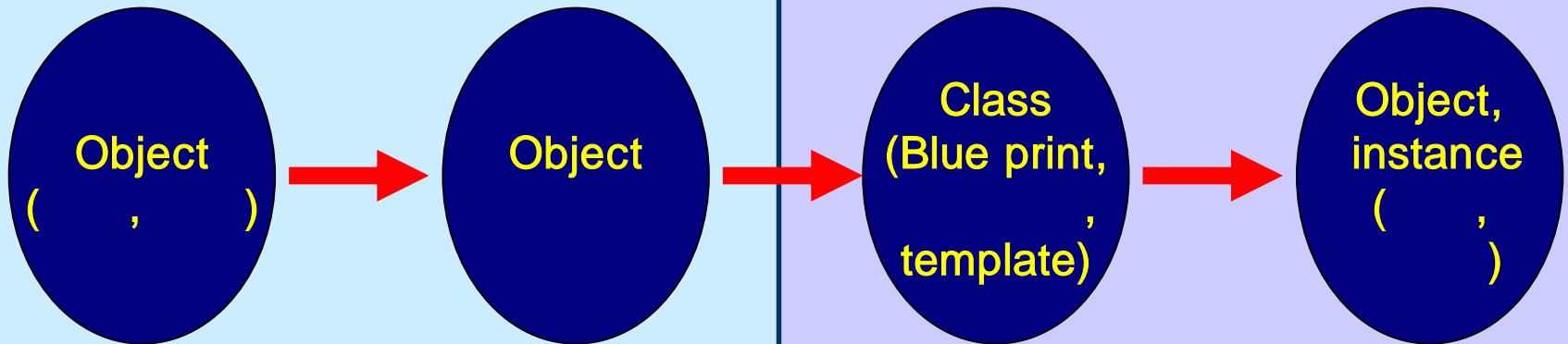
(Real World)

(Software World)

OOA/ Modeling

Abstraction

Instantiation



@

&

3

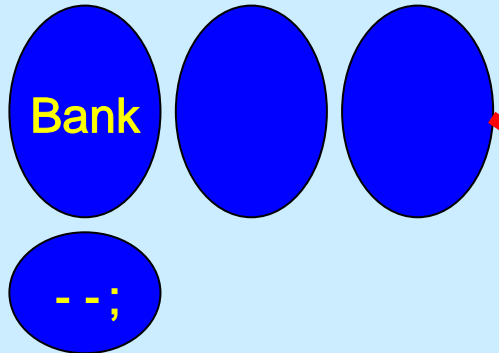
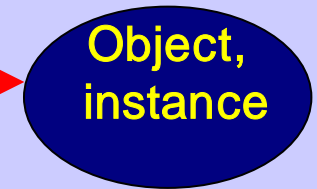
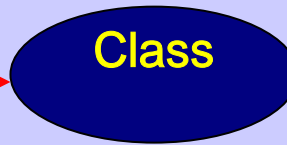
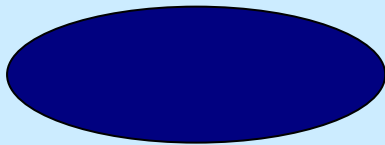
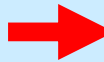
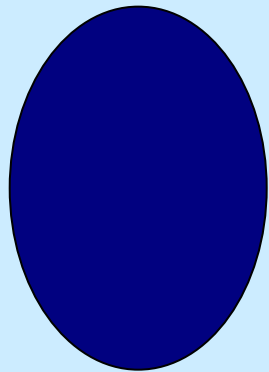
(Real World)

(Software World)

OOA/ Modeling

Abstraction

Instantiation



```
class Account{  
  double money;  
  getMoney(){  
  deposit(){  
  withdraw(){  
  }  
}
```

```
Account acc=  
new Account()
```

@

&

4

Real World

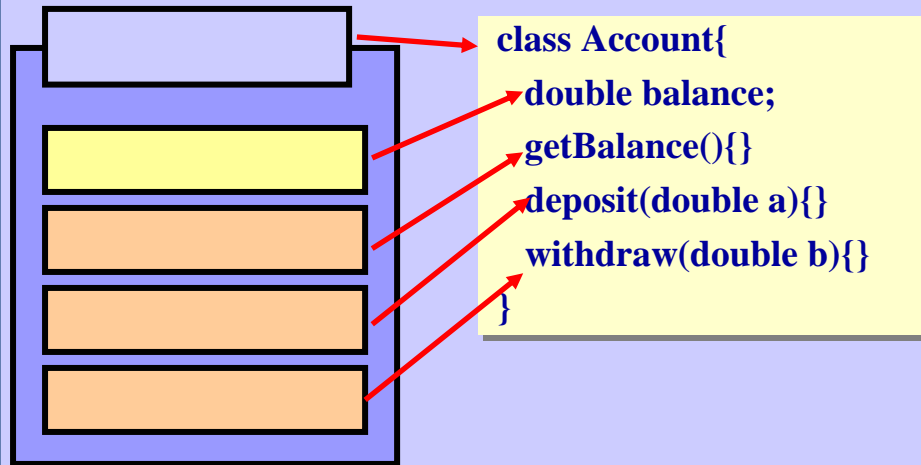
(Software World)

Abstraction

Instantiation

Class

Object,  
instance

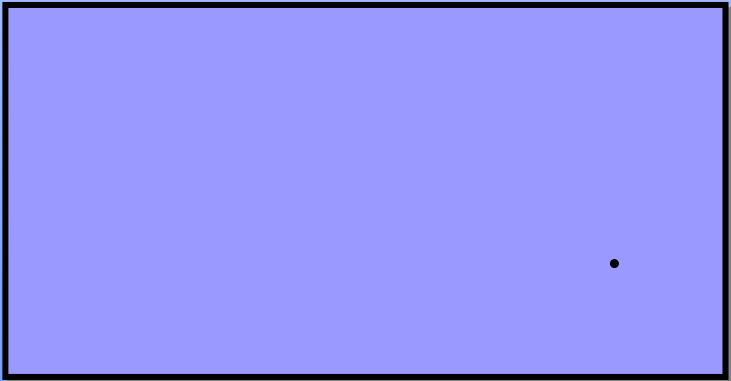


@

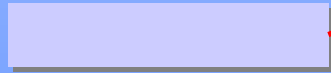
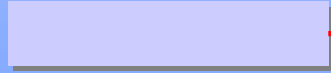
1



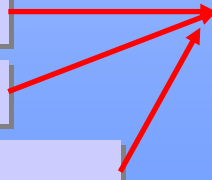
가



**Behavior**



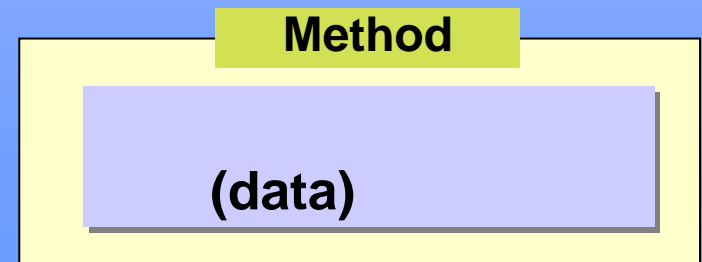
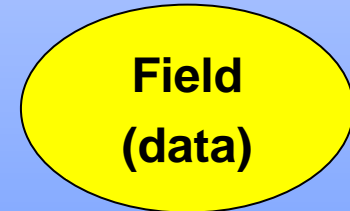
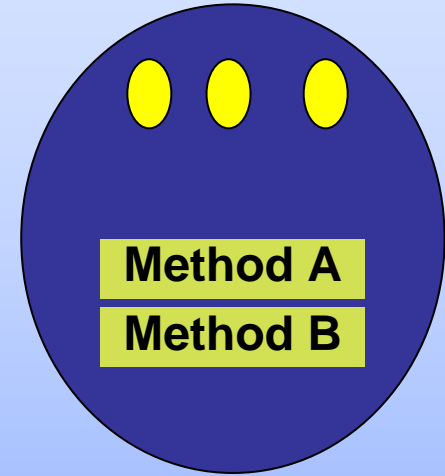
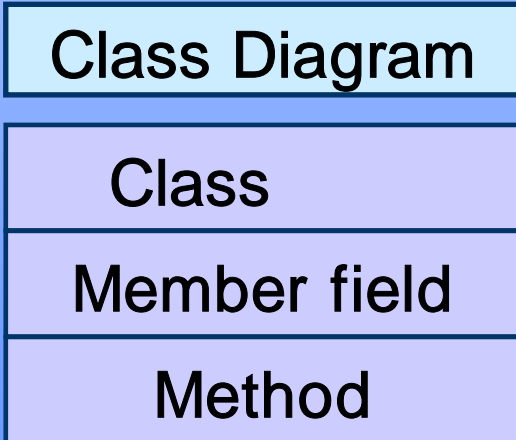
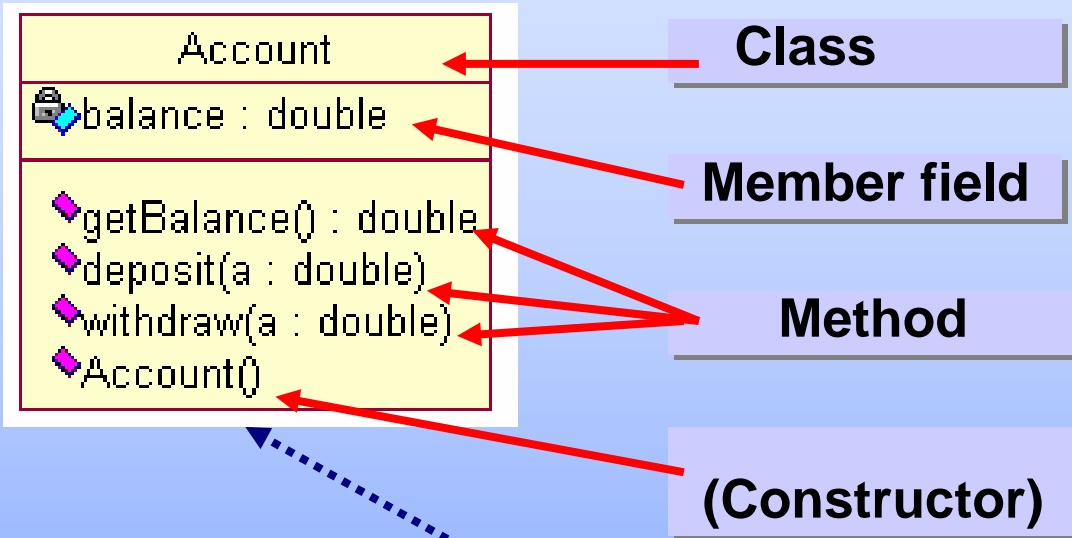
**Attribute**





@

2



## @ Class

- Class

```
[ classModifier ] class ClassName //header
{                                     // body
    /* field      */
    /* method     */
}
```

– classModifier : **public, friend, final, abstract**

@

(1)

- ?  
class, field, method

- private : class 가
- friend( package ) : package(Windows folder ) 가
- protected : package class 가
- public : 가

@

(2)

- final

class

) Java Math header

public final class Math

- abstract

@

- ( field )

:

- 

[ ] field1, field2, ...;

( private,  
package, protected, public ) final,  
static, volatile, transient .

# @ Instantiation

Heap

```
Account acc1=new Account();
```

```
Account acc2=new Account();
```

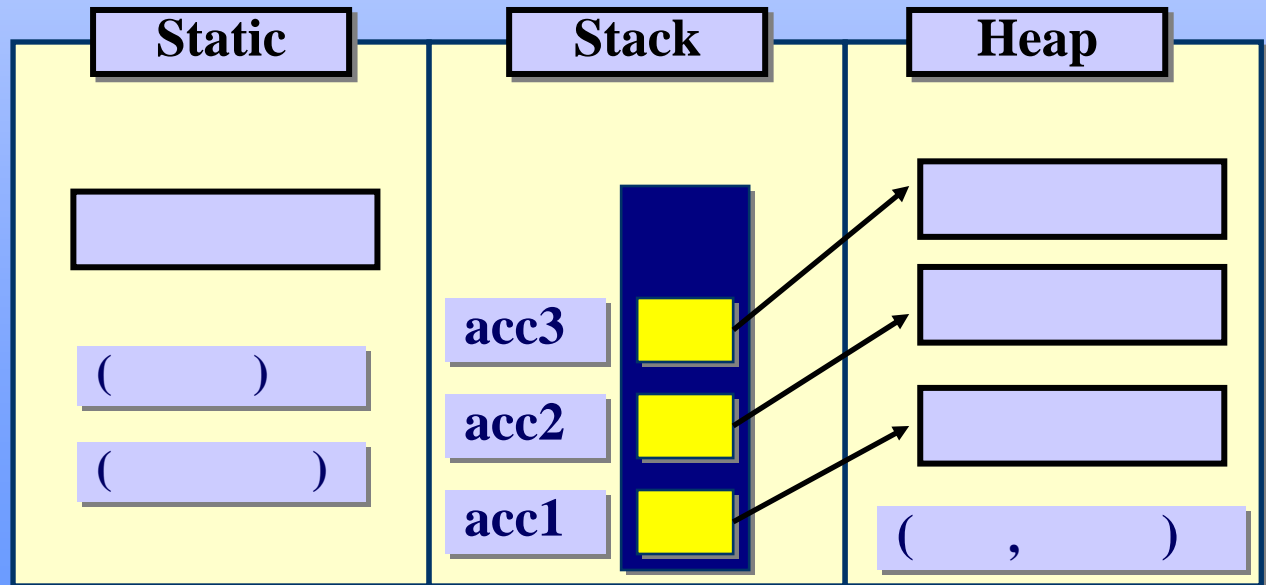
```
Account acc1=new Account();
```

Class

Reference variable

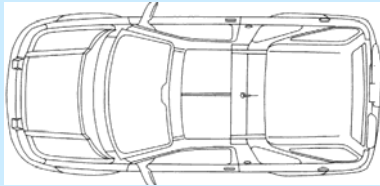
```
Account acc1=new Account();
```

Instance



# @ Instantiation 2

Static



Class

Stack

car3

car2

car1

Heap

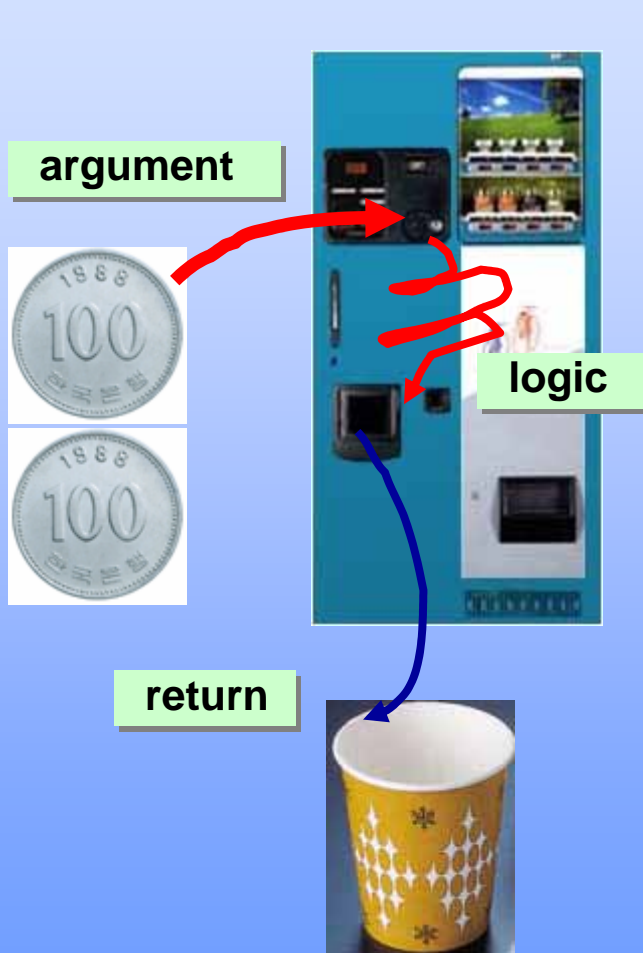


```
Car car1=new Car();
```

```
Car car2=new Car();
```

```
Car car3=new Car();
```

# @ Method



200

```
public int coffee( int money ) {  
    //logic  
    if( money >= 1000 )  
        return 5; // 5  
    else if( money >= 800 )  
        return 4; // 4  
    else - - - ;  
}
```

```
int cups =  
    vendingMachine.coffee( 800 );
```

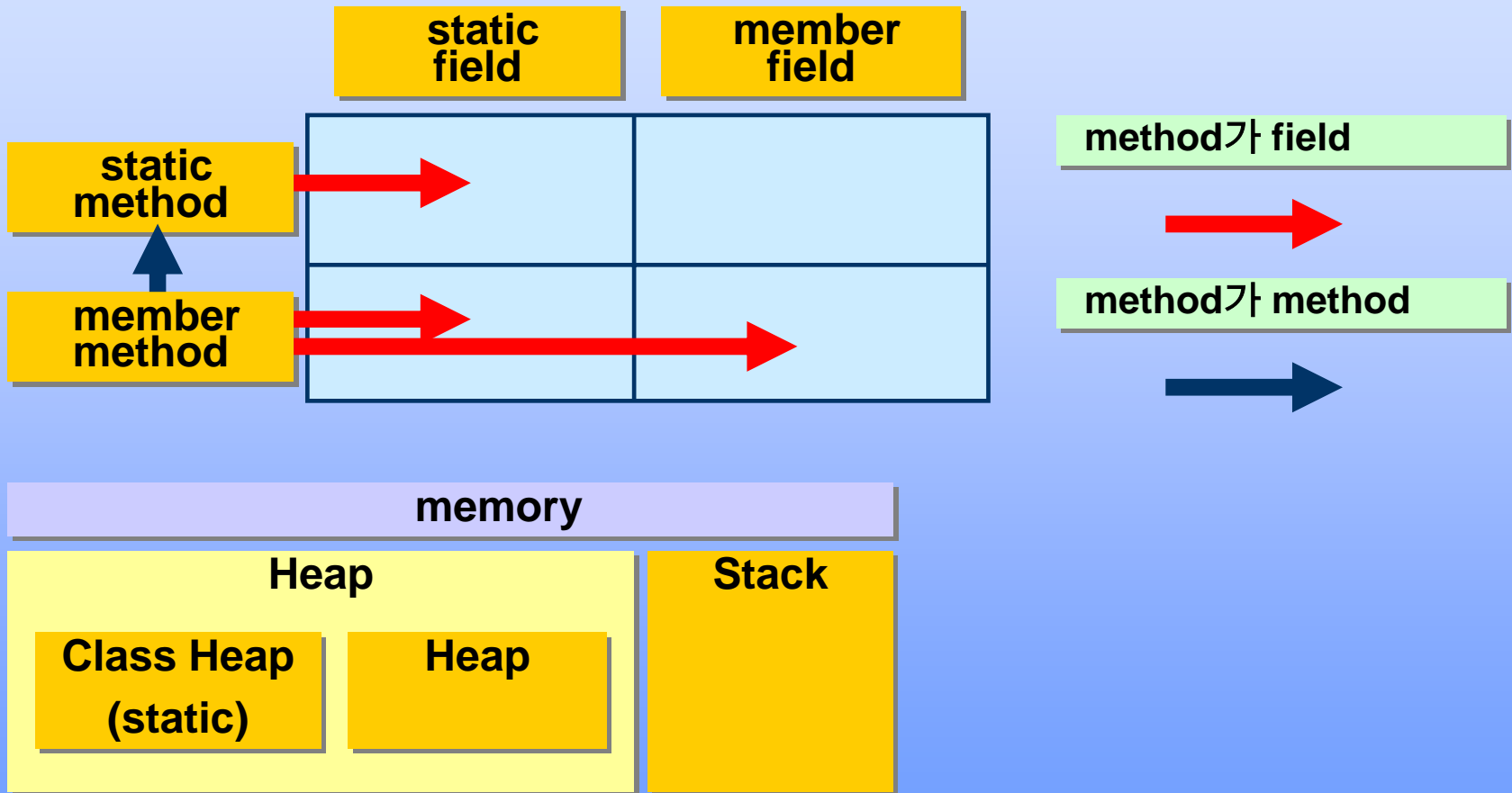
200

→

cups : 4

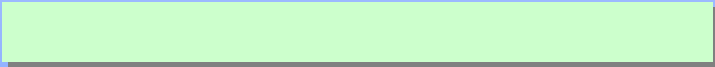
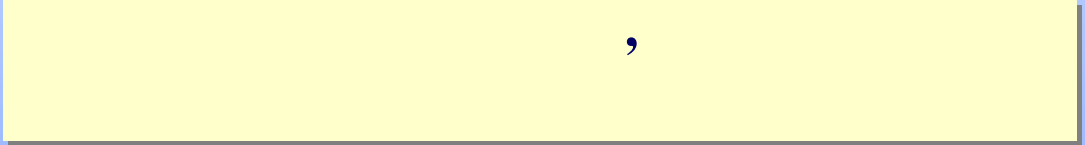
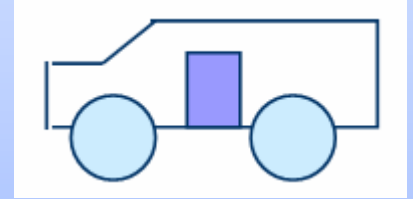
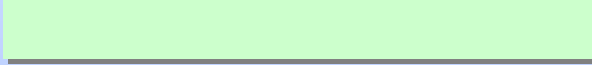


# @ static & non static



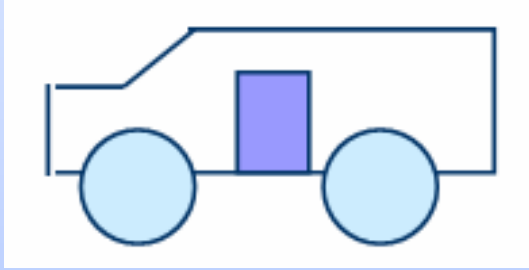
@

3



@

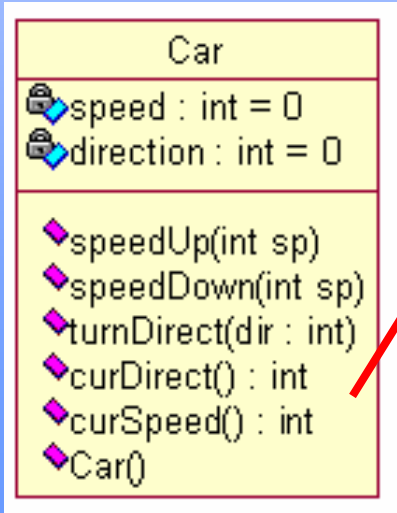
4



: Car

: , ,

: ,  
,



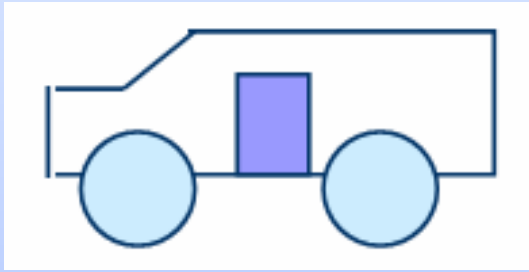
```

public class Car
{
  /**
   *
   */
  private int speed = 0;
  /**
   *      -90      90      0
   */
  private int direction = 0;
  /**
   *      Car default
   */
  public Car(){} //constructor
  /**
   *      5Km/hr      가
   */
  public void speedUp() { speed+=5; }
}

```

@

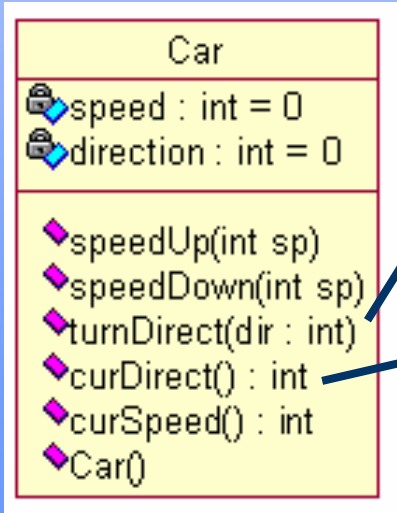
5



: Car

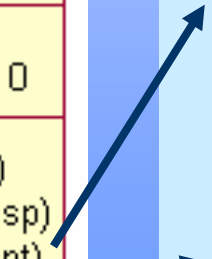
: , ,

: ,  
,



```

/**
     * 5Km/hr
     */
    public void speedDown() { speed-=5; }
    /**
     *
     * (Km/hr)
     * @return int
     */
    public int curSpeed(){ return speed; }
    /**
     *
     *
     * @param dir
     */
    public void turnDirect(int dir) { direction+=dir; }
    /**
     *
     * (0: , -90 , 90 )
     * @return int
     */
    public int curDirect(){ return speed; }
}
  
```



@

# javadoc

**javadoc -use -private ClassName.java**

```

/**
 *
 * @param dir
 */
public void turnDirect(int dir) { direction+=dir; }
/**
 * (0: , -90 , 90 )
 * @return int
 */
public int curDirect(){ return speed; }

```

## Field Summary

private int	<b>direction</b> 오른쪽 -90 왼쪽 90 앞으로 0
private int	<b>speed</b> 자동차의 속도

## Constructor Summary

<b>Car()</b>	자동차 Car default 생성자
--------------	---------------------

## Method Summary

int	<b>curDirect()</b> 자동차의 현재 방향(0: 앞, -90 오른쪽, 90 왼쪽) 반환
int	<b>curSpeed()</b> 자동차의 현재 속도(Km/hr) 반환
void	<b>speedDown()</b> 자동차의 속도를 5Km/hr씩 감소
void	<b>speedUp()</b> 자동차의 속도를 5Km/hr씩 증가
void	<b>turnDirect(int dir)</b> 진행방향을 원하는 각도 만큼 돌린다.

주소(D) C:\javauser\chap4\CarDoc\index.html

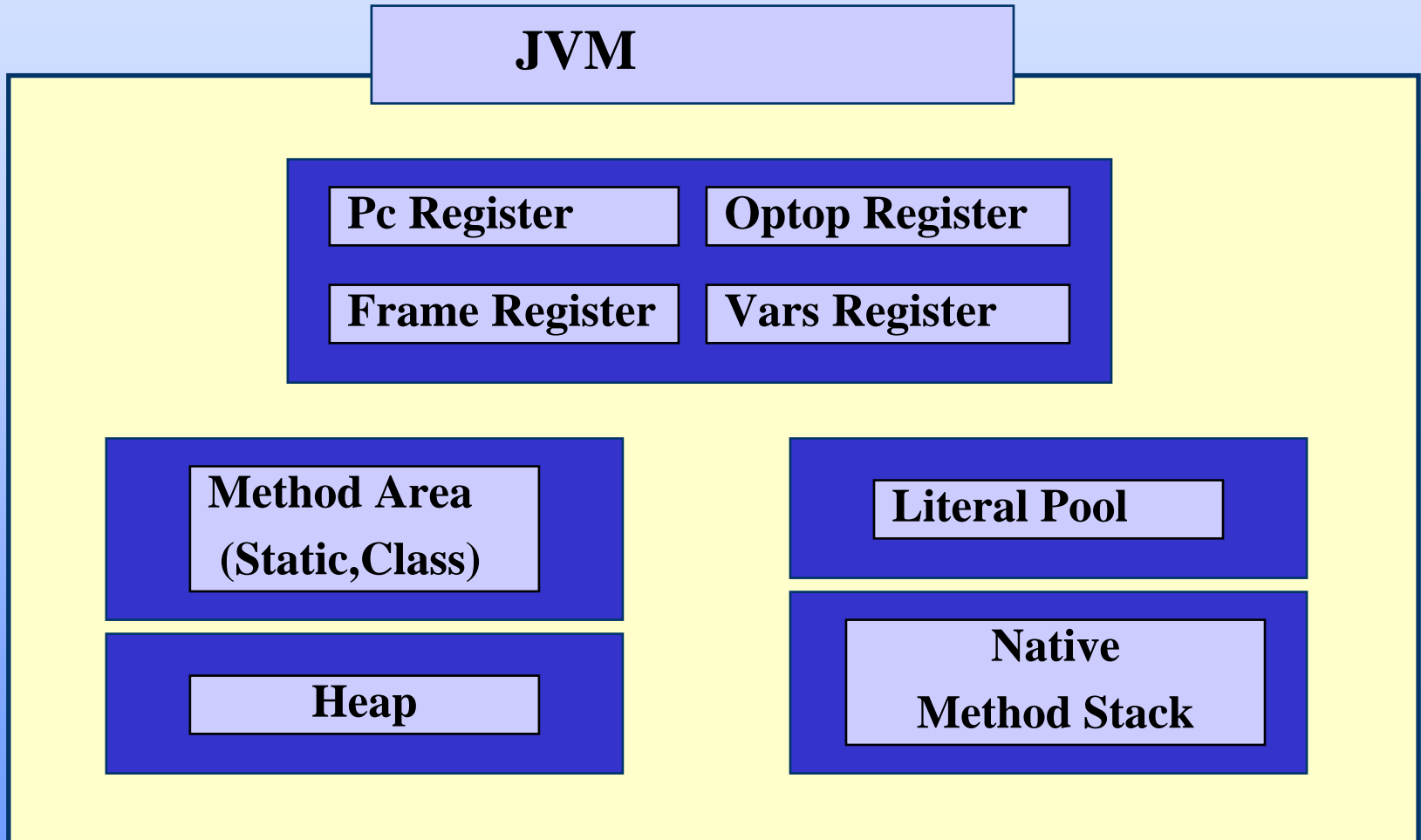
All Classes  
[Car](#)

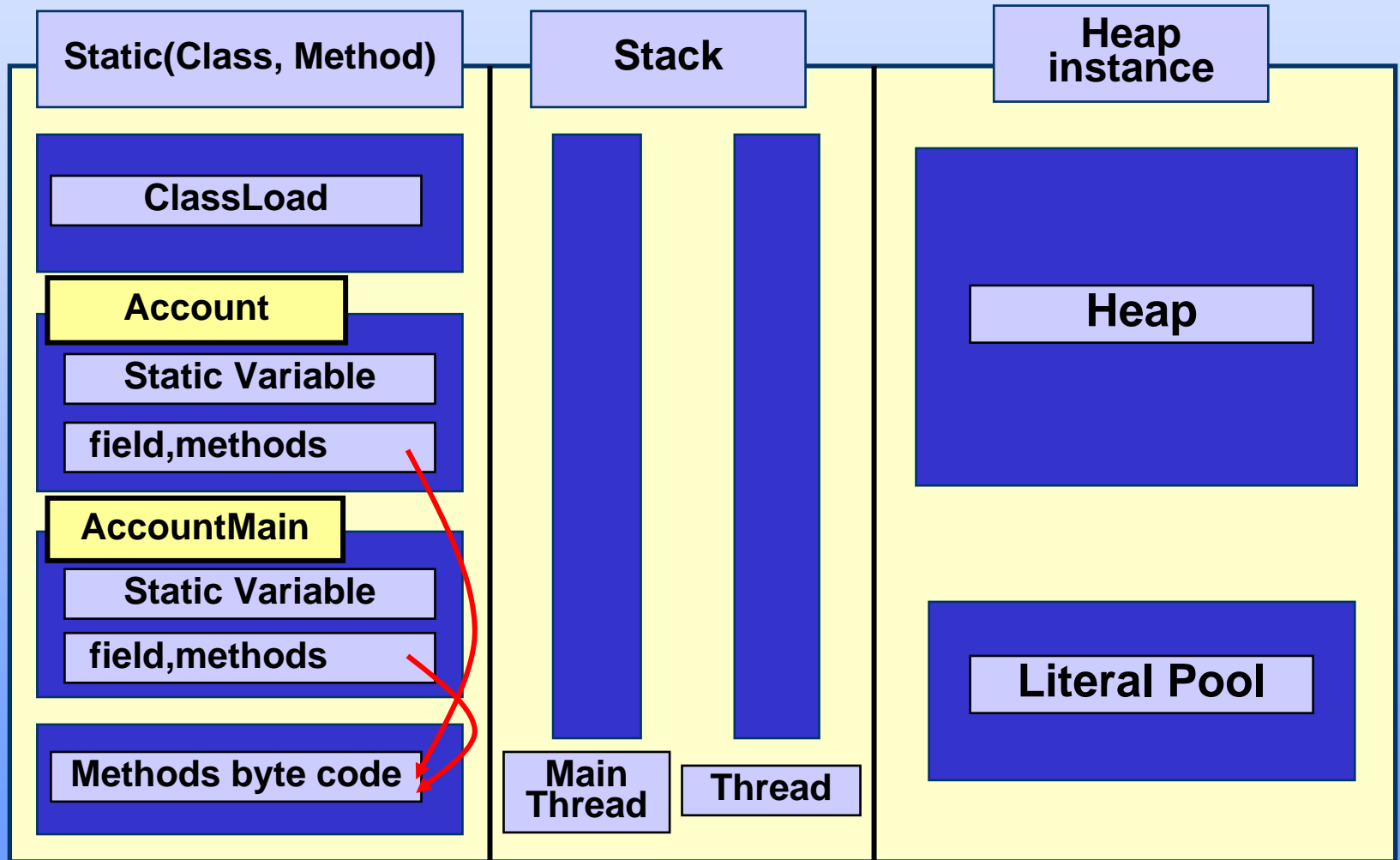
Package **Class** [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)  
 PREV CLASS NEXT CLASS  
 SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

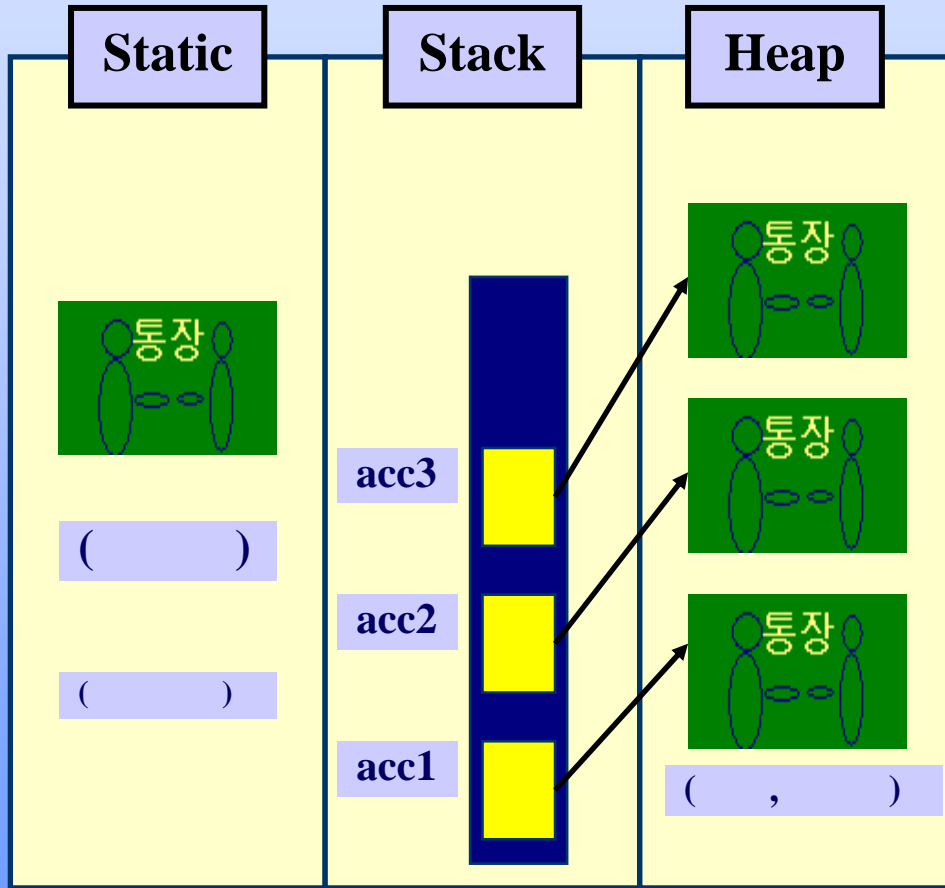
**Class Car**  
 java.lang.Object  
 └ Car

@ JVM

1







### Static

, Class  
static

### Stack

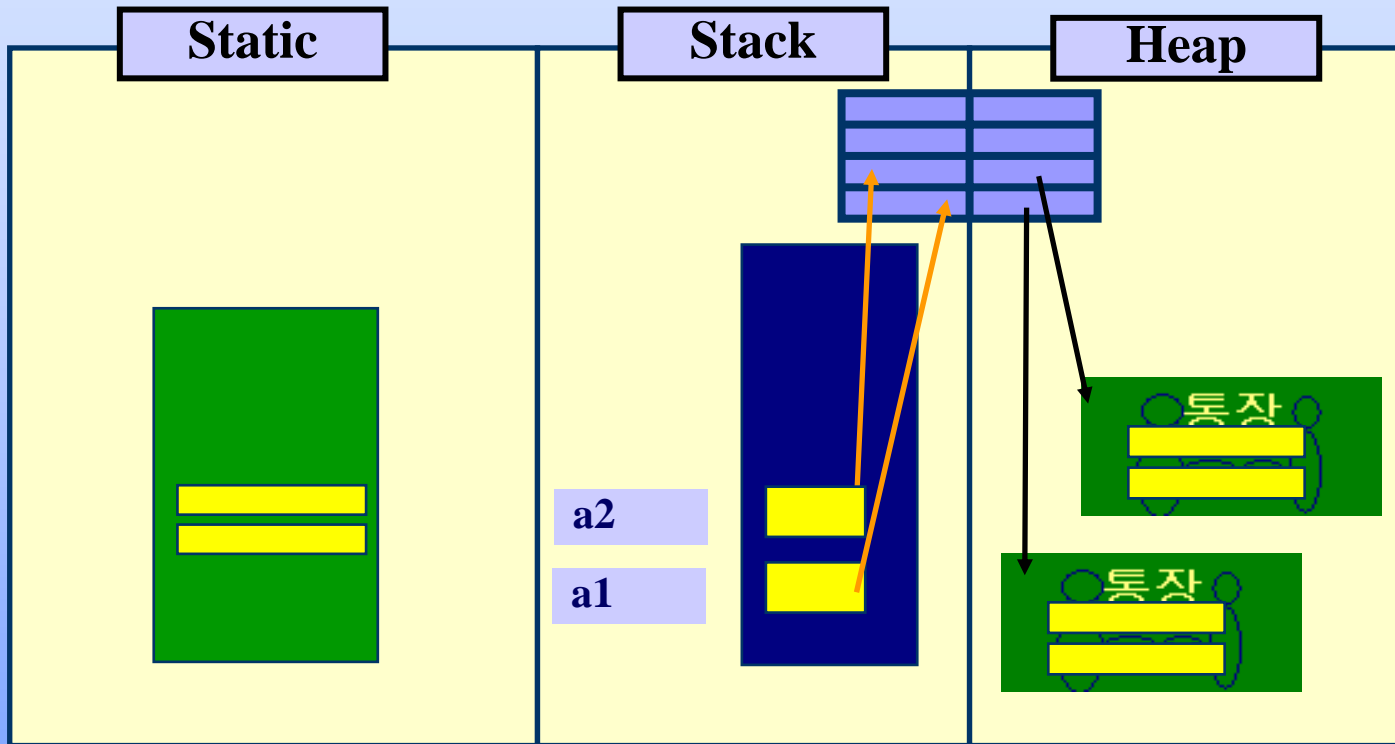
reference  
,  
referencing

### Heap

가  
가



# @ reference pointer



```
Account a1=new Account();  
Account a2=new Account();
```

@

1

```
public class Accounts
{
    private double money=500;    //
    public Accounts ( double money ){ //
        this.money=money;}
    public double getMoney(){    //
        return money;}
    public void withdraw(double amount){ //
        if((amount>0)&&(money-amount>=0)){
            money-=amount;}}
    public void deposit(double amount){ //
        if(amount>0){
            money+=amount;}}
}
```

@

2

```
public class AccountMain  
{  
    public static void main( String[] args)  
    {  
        Accounts acc1=new Accounts(1000); //  
        acc1.deposit(3000); //  
        acc1.deposit(2000); //  
        acc1.withdraw(500);//  
        System.out.println(acc1.getMoney());//  
    }  
}
```

@

3

```
public class Student {  
    String name;  
    int classes;  
    int num;  
}
```

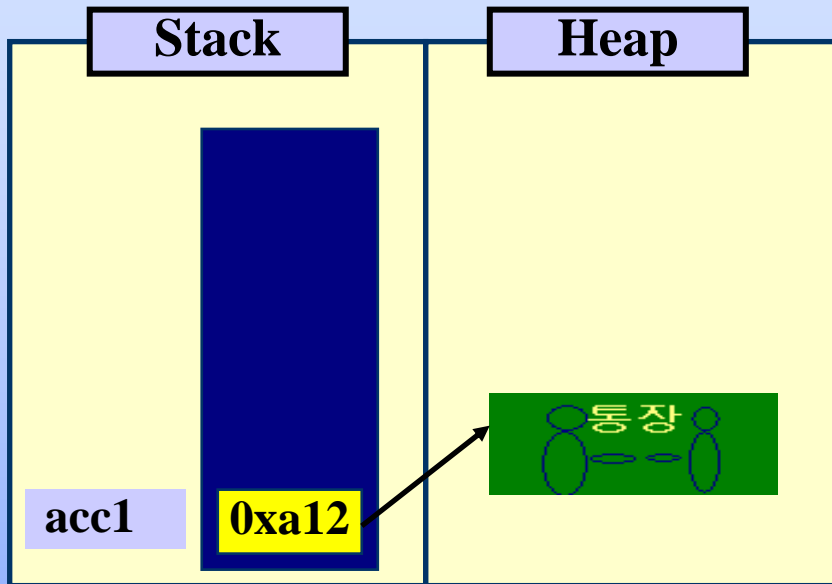
```
//main
```

```
Student stu1=new Student();  
Student stu2=new Student();  
Student stu3=new Student();
```

```
stu1.name="ToTo"; stu1.classes=3; stu1.num=14;  
stu2.name="KaKa"; stu2.classes=2; stu2.num=11;  
stu3.name="YaYa"; stu3.classes=1; stu3.num=9;
```

```
System.out.println(stu1.name);  
System.out.println(stu2.name);  
System.out.println(stu3.name);
```

@



```
public class Account{  
    private double money=500;  
    public Account(double money){  
        this.money=money;  
    }  
}
```

**Field**



**Field**



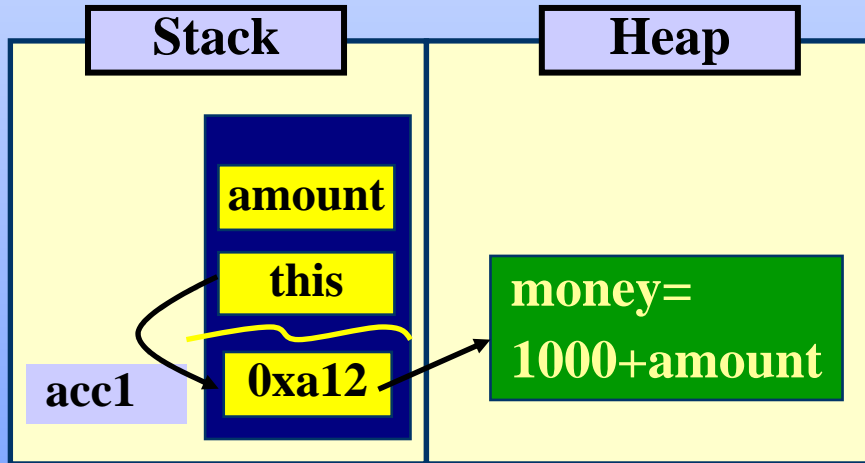
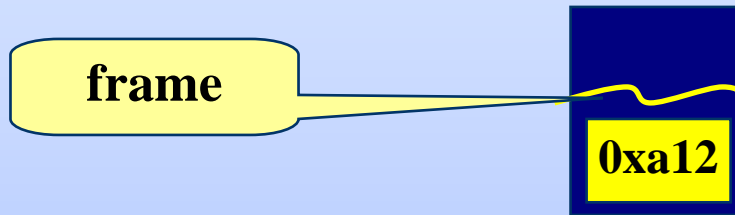
**Constructor**

**Field**

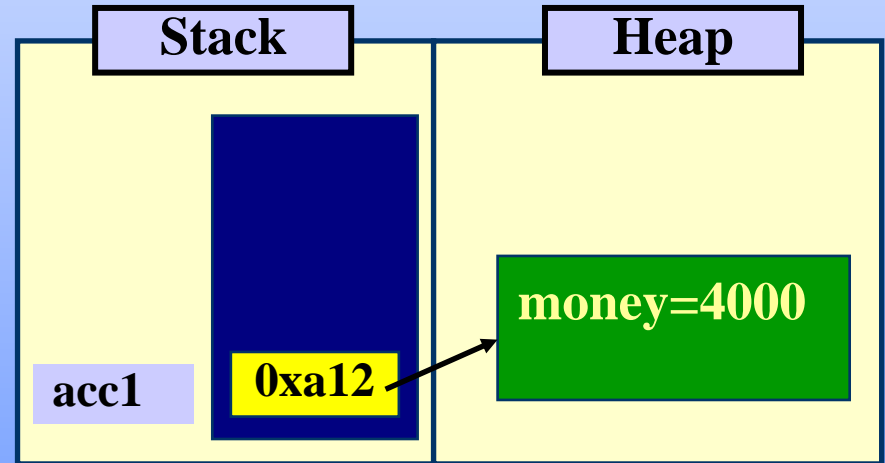


**Reference Variable**



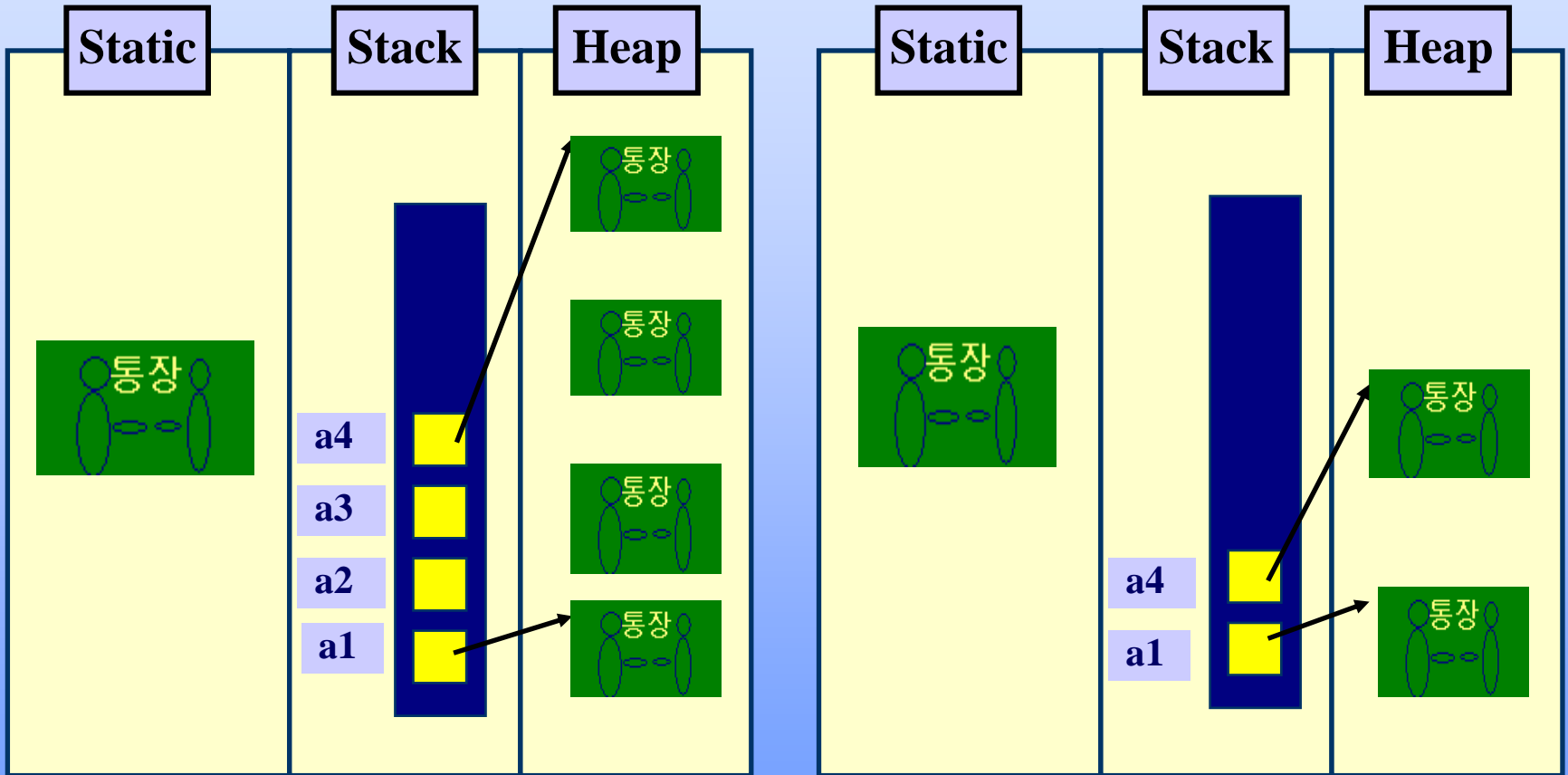


```
Accounts acc1=new Accounts(1000);  
acc1.deposit(3000);
```



```
method가  
amount, this, frame  
stack
```

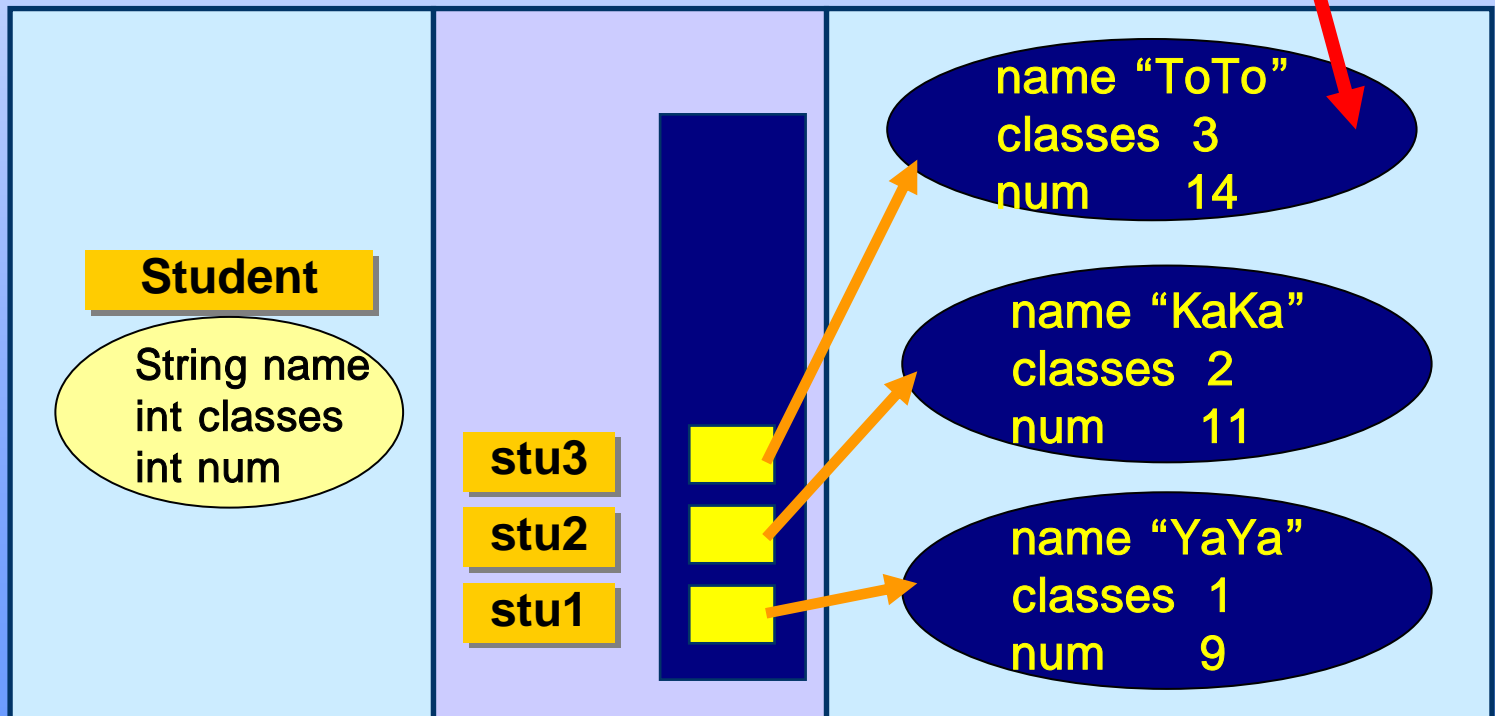
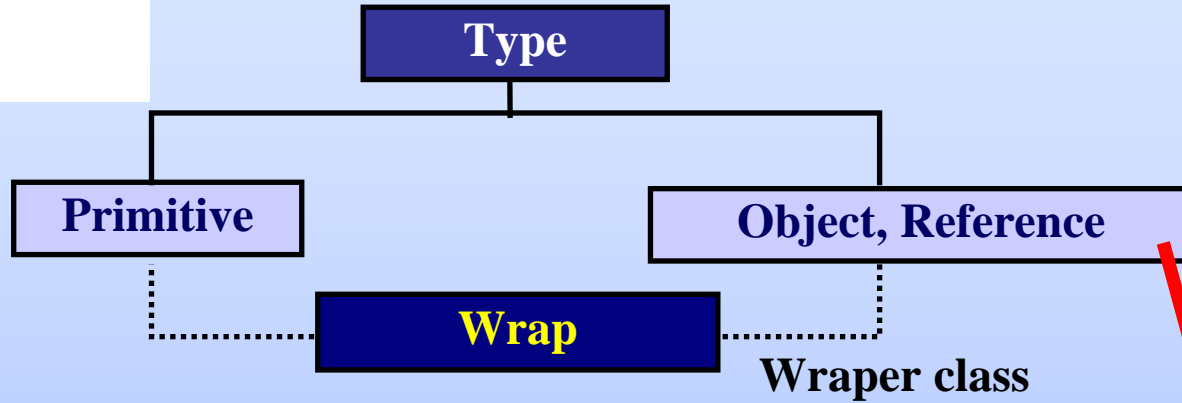
# @ Garbage Collection



a2=null; // a2 → Garbage Collection

a3=null; // a3 → Garbage Collection

@ Type





# @ Airplane – , , 1

```
public class Airplane {  
  
    String nameOfCo="Corea Air"; //  
    String nameOfAirp="C10111"; //  
    int passenger=87;           //  
    double weightOfLu=0.614;   //Kg  
    int fuel=10000;             //liter  
    int goPerL=10;              //Km/liter  
  
    public Airplane(){}        //  
  
}
```

# @ Airplane – , , 2

```
public class AirplaneMain {
    public static void main(String[] args) {
        Airplane air858=new Airplane();
        air858.nameOfAirp="                ";//
        air858.nameOfCo="                "; //
        air858.passenger=200;           //
        air858.fuel=6000;                //liter
        air858.goPerL=11;                //1km/liter
        air858.weightOfLu=0.4;          //kg/liter

        System.out.print("                : "+air858.passenger);
        System.out.print("                : "+air858.nameOfCo);
        System.out.print("                : "+air858.nameOfAirp);
        System.out.print("                : "+air858.goPerL*air858.fuel);
        System.out.println("                : "+air858.fuel*air858.weightOfLu);
    }
}
```

```
public class Airplane1 {
    private String nameOfCo="Corea Air"; //
    private String nameOfAirp="C10111"; //
    private int passenger=87;           //
    private double weightOfLu=0.614;   //Kg
    private int fuel=10000;             //liter
    private int goPerL=10;              //Km/liter

    public int getFuel() {
        return fuel;
    }
    public int getGoPerL() {
        return goPerL;
    }
    public String getNameOfAirp() {
        return nameOfAirp;
    }
    public String getNameOfCo() {
        return nameOfCo;
    }
}
```

## @ Airplane – , , 4

```
public int getPassenger() {
    return passenger;
}
public double getWeightOfLu() {
    return weightOfLu;
}
public void setFuel(int i) {
    fuel = i;
}

public void setGoPerL(int i) {
    goPerL = i;
}
public void setNameOfAirp(String string) {
    nameOfAirp = string;
}
public void setNameOfCo(String string) {
    nameOfCo = string;
}
```

# @ Airplane – , , 5

```
public void setPassenger(int i) {
    passenger = i;
}

public void setWeightOfLu(double d) {
    weightOfLu = d;
}

public String about(){
    String s="                : "+passenger;
    s+="                : "+nameOfCo;
    s+="                : "+nameOfAirp;
    s+="                : "+goPerL*fuel;
    s+="                : "+fuel*weightOfLu;
    return s;
}
}
```

# @ Airplane – , , 6

```
public class Airplane1Main {  
  
    public static void main(String[] args) {  
  
        Airplane1 air849 =new Airplane1();  
        air849.setFuel(6000);  
        air849.setGoPerL(11);  
        air849.setNameOfAirp("        ");  
        air849.setNameOfCo("        ");  
        air849.setPassenger(200);  
        air849.setWeightOfLu(0.4);  
        System.out.println(air849.about());  
  
    }  
}
```

# @ Airplane – , , 7

```
public class Airplane2 {  
    private int goPerL=10;           //Km/liter  
    private boolean isReady=false;  
    ...  
    public void setFuel(int i) {  
        fuel = i;  
        if(i<0){  
            fuel=0;  
            isReady=false;  
        }else if(i>0 && i<5000){  
            fuel=i;  
            isReady=false;  
        }else {  
            isReady=true;  
        }  
    }  
    ...  
}
```

# @ Airplane – , , 8

```
public class Airplane3 {
    ...
    //setXXX method
    public Airplane3(String a,String b,int c,double d,int e,int f){
        nameOfCo    = a;    //
        nameOfAirp  = b;    //
        passenger    = c;    //
        weightOfLu  = d;    //Kg/liter
        fuel         = e;    //liter
        goPerL      = f;    //km/liter
    }
}

public class Airplane3Main {

    public static void main(String[] args) {

        Airplane3 air949 =new Airplane3("    ",
        //      --
        System.out.println(air949.about());
    }
}
```



```
public class Airplane4 {
    ....
    // referece type
    public String toString(){
        String s="                : "+passenger;
        s+="                : "+nameOfCo;
        s+="                : "+nameOfAirp;
        s+="                : "+goPerL*fuel;
        s+="                : "+fuel*weightOfLu;
        return s;
    }
}

public static void main(String[] args) {

    Airplane4 air949 =new Airplane4("                ", "                ",200,0.4,6000,11);
    //                - -
    System.out.println(air949);
    System.out.println("=====");
    System.out.println(air949.toString());
}
```

```
public class Airplane5 {  
    ....  
    public Airplane5(String nameOfCo,String nameOfAirp,  
        int passenger,double weightOfLu,  
        int fuel,int goPerL){  
        this.nameOfCo    = nameOfCo;    //  
        this.nameOfAirp  = nameOfAirp;  //  
        this.passenger   = passenger;   //  
        this.weightOfLu  = weightOfLu;  //Kg/liter  
        this.fuel        = fuel;        //liter  
        this.goPerL     = goPerL;      //km/liter  
    }  
    public Airplane5(String nameOfCo,String nameOfAirp,  
        int passenger,int fuel){  
        this.nameOfCo    = nameOfCo;    //  
        this.nameOfAirp  = nameOfAirp;  //  
        this.passenger   = passenger;   //  
        this.fuel        = fuel;        //liter  
    }  
}
```

# @ Airplane – , , 11

```
public class Airplane5Main {  
  
    public static void main(String[] args) {  
        Airplane5 air948 =new Airplane5("        ","  
            ",200,0.4,6000,11);  
        //        - -  
        System.out.println(air948);  
        System.out.println(air948.toString());  
        System.out.println("=====");  
        Airplane5 air947 =  
            new Airplane5("        ","        ",300,7000);  
        System.out.println(air947);  
    }  
}
```

# @ Computer Systems

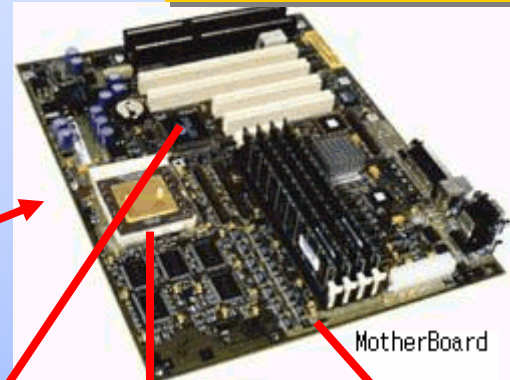
Computer

Monitor



Body

MotherBoard



CPU

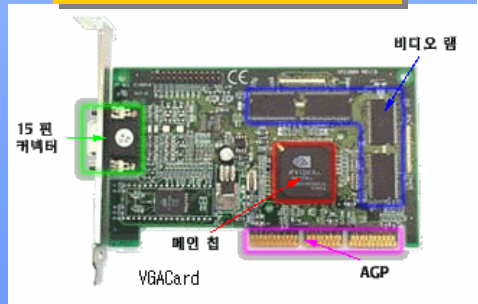


SoundCard



KeyBoard/Mouse

VGACard



@

1

```
public class Computer {
    String name="XG2 Long Canvas";
    Moniter monitor=new Moniter();
    Speaker speaker1=new Speaker();
    Speaker speaker2=new Speaker();
    Mouse mouse=new Mouse();
    KeyBoard keyBoard=new KeyBoard();
    MotherBoard motherBoard=new MotherBoard();
}

class Monitor{
    int size=17; //inch
}

class Speaker{
    int volume=10; //0~20 db
}
```

@

2

```
class Mouse{
    String type=" ";
}
class KeyBoard{
    int keys=103;
}
class MotherBoard{
    CPU cpu=new CPU();
    SoundCard soundCard=new SoundCard();
    VGACard vGACard=new VGACard();
}
class CPU{
    int speed=1700;//khz
}
class SoundCard{
    String company=" ";
}
class VGACard{
    int color =20000;//
}
```

@

3

```
public class ComputerMain {  
  
    public static void main(String[] args) {  
        Computer com=new Computer();  
  
        System.out.println("          : "+com.name);  
        System.out.println("          : "+com.monitor.size);  
        System.out.println("CPU          : "+com.motherBoard.cpu.speed);  
        System.out.println("          : "+com.motherBoard.soundCard.company);  
        System.out.println("VGA          : "+com.motherBoard.vGACard.color);  
  
    }  
}
```

## @ Array

```
int a=3;  
int b=5;  
Student ss=new Student();
```

```
int [ ] nums;  
Student [ ] stus;  
nums=new int[5];
```

```
int [ ] num=new int[5];  
Student [ ] stu=new Student[5];
```

```
int[0]=3;  
stu[0]=new Student();
```



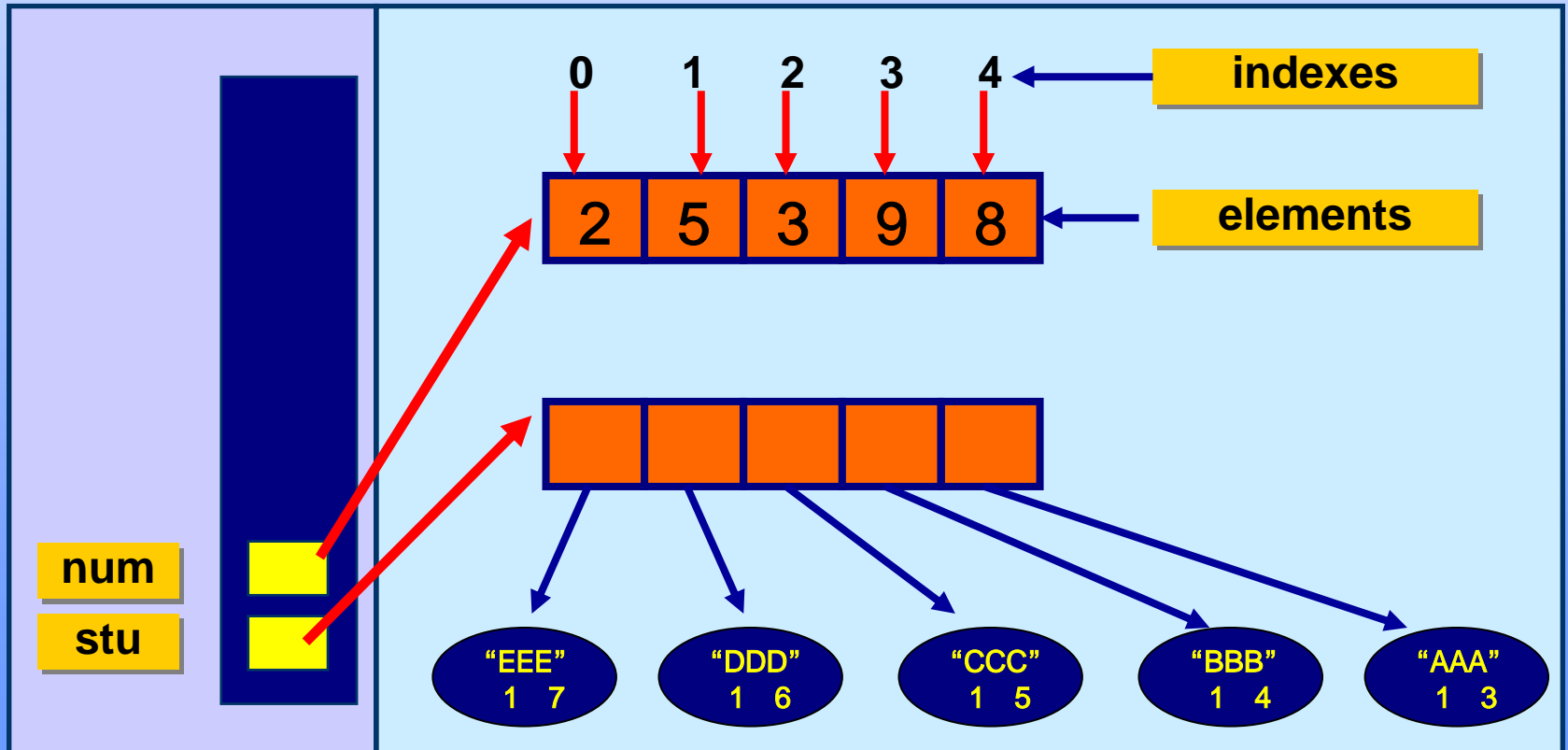
# @ Array

```
Student [ ] stu=new Student[5];
```

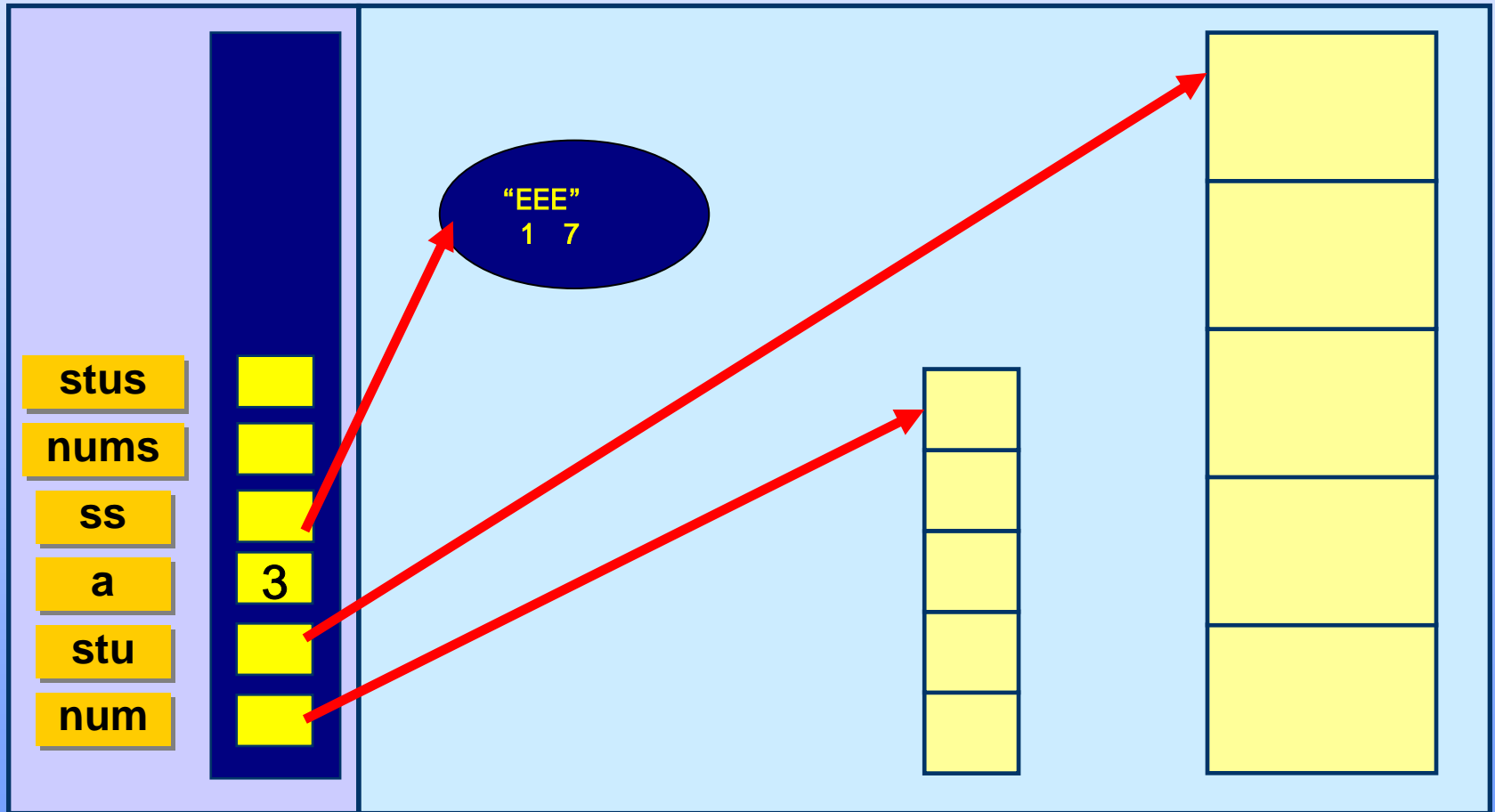
```
...
```

```
int [ ] num=new int[5];
```

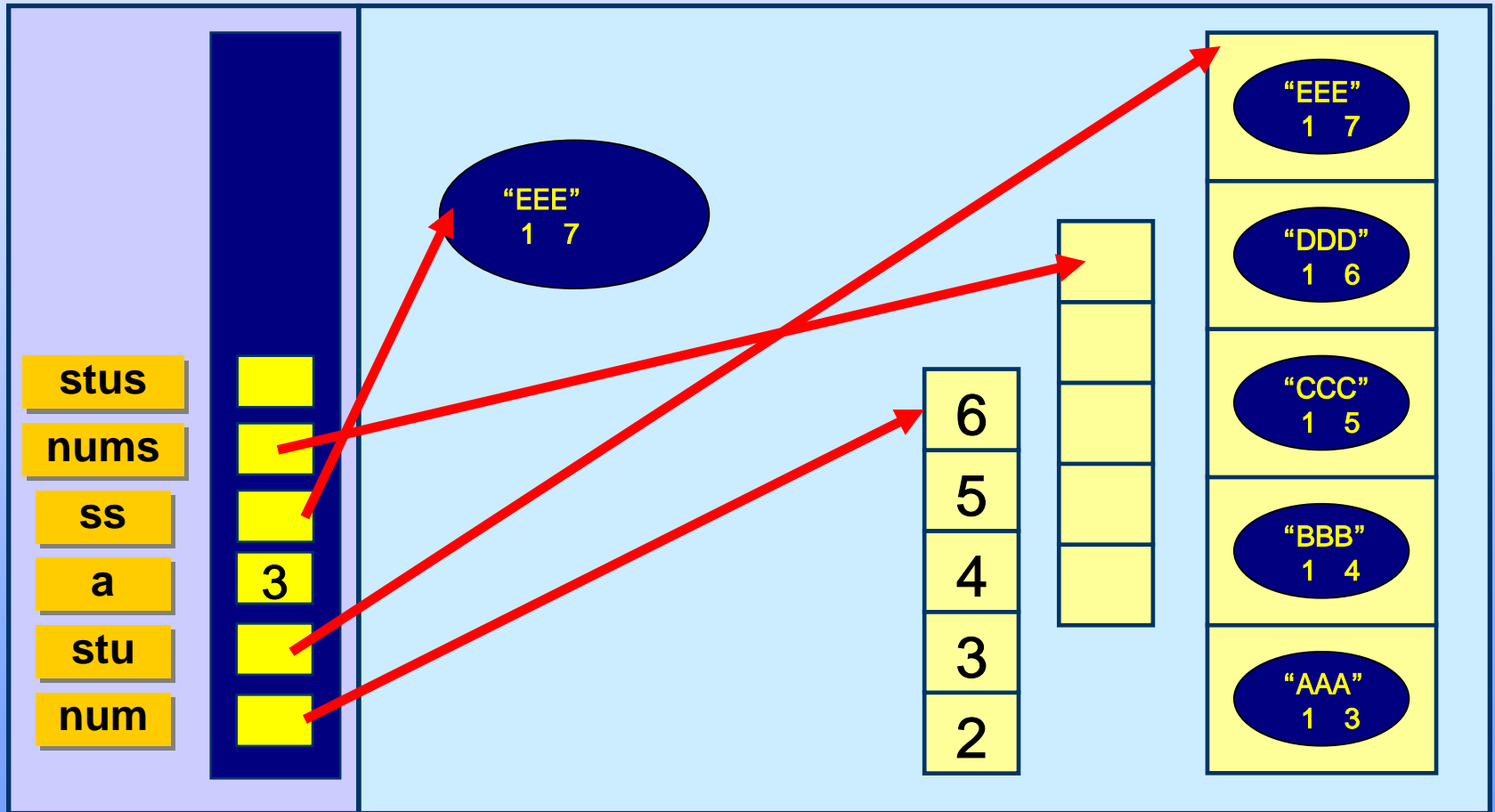
```
...
```



# @ Array Declaration/Define



# @ Array Initialization



# @ Array-

# 가

# 1

```
public class BusNon {  
    int people=1;        //  
    int amountOfOil=200; // 200 liter  
    int usedOil=0;      // 200 liter  
    String start="      ";  
    String end="        ";  
    String currStop;  
    int totalDis;  
  
    public final static int GOPEROIL = 5; // 5 liter/km  
}
```

## @ Array-

## 가

## 2

```
public class BusNonM {
    public static void main(String[] args) {
        String [] busStop={"", "", "", "", "", ""};
        int [] distance={0,5,8,7,5,6};
        int [] passsengerIn={5,18,20,15,20,0};
        int [] passsengerOut={0,3,6,10,6,0}; //
        int totalIn=0;
        int totalOut=0;

        BusNon bus85=new BusNon();
        for(int i=0;i<passsengerIn.length;i++){
            totalIn+=passsengerIn[i];
        }

        System.out.println(" : "+totalIn);
        for(int i=0;i<passsengerOut.length - 1;i++){
            totalOut+=passsengerOut[i];
        }
    }
}
```

```
passengerOut[passengerOut.length - 1]=totalIn - totalOut;
System.out.println("                : “
                +passengerOut[passengerOut.length - 1]);
for(int i=0;i<busStop.length;i++){
    bus85.currStop=busStop[i];
    bus85.people -=passengerOut[i];
    bus85.people+=passengerIn[i];
    bus85.totalDis+=distance[i];
    bus85.uesedOil+=distance[i]*BusNon.GOPEROIL;
    bus85.amountOfOil=200 - bus85.uesedOil;
    System.out.print("                :"+bus85.currStop+
    " t                : "+bus85.totalDis);
    System.out.print(" t                : "+bus85.people+
    " t                :"+bus85.amountOfOil);
    System.out.println(" t                : "+bus85.uesedOil);
}
}
}
```

# @ Array-

# 가

# 1

```
public class Bus {
// memeber field
private int people=1;      //
private int amountOfOil=200; // 200 liter
private int  usedOil=0;    // 200 liter
private String start="    "; //
private String end="      "; //
private String currStop;   //
private int totalDis;      //          Km
private int totalIn;      //
private int totalOut;     //
//
public final static int GOPEROIL = 5; // 5 liter/km

//          가?
public int getAmountOfOil() {
    return amountOfOil;
}

public String getEnd() {
    return end;
}
```

```
public int getPeople() {
    return people;
}
public String getStart() {
    return start;
}
public String getCurStop() {
    return currStop;
}
public void setCurStop(String curs) {
    currStop=curs;
}
public void useOfOil(int i) {
    amountOfOil -= i*Bus.GOPEROIL;
    usedOil+=i*Bus.GOPEROIL;
}
public void inPeople(int i) {
    people+=i;
}
public void outPeople(int i) {
    people -= i;
}
public int getTotalDis() {
    return totalDis;
}
```



## @ Array-

## 가

## 3

```
public void goDis(int i) {
    totalDis += i;
}
public String toString(){
    return "      :"+currStop+" t      : "+totalDis+" t      : "+people+" t
    :"+getAmountOfOil()+" t      : "+uesedOil;
}
public int getTotalIn() {
    return totalIn;
}
public int getTotalOut() {
    return totalOut;
}
public void setTotalIn(int i) {
    totalIn += i;
}
public void setTotalOut(int i) {
    totalOut += i;
}
```

# @ Array-

# 가

# 4

```
public class BusCM{
public static void main(String[] args) {
String [] busStop={"", "", "", "", "", ""};
int [] distance={0,5,8,7,5,6};
int [] passsengerIn={5,18,20,15,20,0};
int [] passsengerOut={0,3,6,10,6,0}; //
int totalOut=0;
Bus bus85=new Bus();
for(int i=0;i<passsengerIn.length;i++){
    bus85.setTotalIn(passsengerIn[i]);
}
System.out.println(" : "+bus85.getTotalIn());
for(int i=0;i<passsengerOut.length - 1;i++){
    bus85.setTotalOut(passsengerOut[i]);
}
System.out.println(" :”
+(bus85.getTotalIn() - bus85.getTotalOut()));
}
}
```

# @ Array-

# 가

# 5

```
public class BusCM{
    public static void main(String[] args) {
        String [] busStop={"", "", "", "", "", ""};
        int [] distance={0,5,8,7,5,6};
        int [] passsengerIn={5,18,20,15,20,0};
        int [] passsengerOut={0,3,6,10,6,0}; //
        int totalOut=0;
        Bus bus85=new Bus();
        for(int i=0;i<passsengerIn.length;i++){
            bus85.setTotalIn(passsengerIn[i]);
        }
        System.out.println(" : "+bus85.getTotalIn());
        for(int i=0;i<passsengerOut.length - 1;i++){
            bus85.setTotalOut(passsengerOut[i]);
        }
        System.out.println(" :”
            +(bus85.getTotalIn() - bus85.getTotalOut()));
    }
}
```

@

, ,

( , )