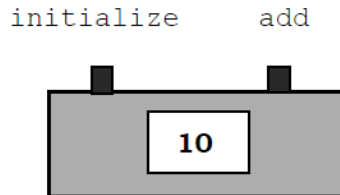


LEC5 OOP 2018-2019

Example1

let us consider the example of counters (المعدادات). A counter is a device that keeps account of the number of times an event has occurred. It has two buttons: an initialize button (زر القيمة الابتدائية) that resets the counter to 0, and an add button (زر الاضافة) that adds 1 to its present number as shown in the following figure a counter with a number 10.

[Grab your reader's attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]



Structurally, the first counter object can be represented as follows:

First Counter Object

Attributes:

number = 10

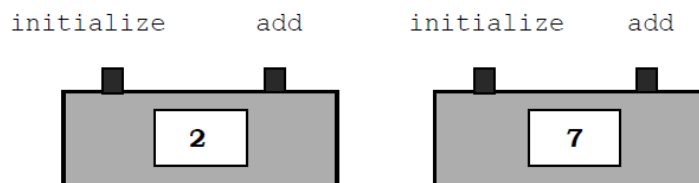
Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

The next figure shows two more counters.



Like the first counter, these two counters may be reset to zero and incremented through the `initialize` and `add` buttons respectively, and represented as follows:

LEC5 OOP 2018-2019

Second Counter Object

Attributes:

number = 2

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

Third Counter Object

Attributes:

number = 7

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

All the three counters share the same definition of attributes and methods, and like in the previous examples, they can be defined by a class as follows:

Class Counter

Attributes:

number

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

The Counter class has:

- an attribute, number;
- an initialize() method that causes a counter to reset its number to 0.
- an add() method that causes a counter to add 1 to its number; and
- a getNumber() method that returns the current value of the attribute number.

LEC5 OOP 2018-2019

Suppose a new object is created from the Counter class. Although the new Counter object would have the same definition of attributes and methods as the previous three counters, its attribute value may not be the same as the other counters. This suggests that the state of the counters may be different from one another.

For the newly created fourth Counter object, it has a state represented by the attribute `number` with a value of 0, the value at initialization:

```
Fourth Counter Object
Attributes:
  number = 0
Methods:
  add()          {number = number + 1}
  initialize()  {number = 0}
  getNumber()   {return number}
```

Note that the attribute value of the fourth Counter object is different from the other three counters.

Creating three objects (counters) using Java language. (H.W.)

Example 2

Trace the following programs and print the output of each one.

I-

```
public class Counter{
private int number;
public void add(){
number=number+1;}
public void initial(){
number=0;}
public int get_number(){
return(number);}
}
public class Main(){
public static void main(String [] args){
Counter c1,c2;
c1=new Counter(); c2=new Counter();
c1. initial ();
c1.add();
```

LEC5 OOP 2018-2019

```
System.out.println(c1.get_number());
//Or int k=c1.get_number();System.out.println(k);
c2.initial();
c2.add();
c2.add();
System.out.println(c2.get_number());
} }
```

II-

```
public class Main(){
public static void main(String [] args){
Counter c1,c2;
c1=new Counter();
c1. initial ();
c1.add();
System.out.println(c1.get_number());
c2. initial (); c2.add(); c2.add();
c2.add();
System.out.println(c2.get_number()); } }
```

III-

```
public class Main(){
public static void main(String [] args){
Counter c1;
c1=new Counter();
c1. initial ();
for(int i=1;i<=3;i++){
c1.add();
System.out.println(c1.get_number());}
//discuss if initial method is inside loop then printing out of loop
}}
```

Note: The order of methods within a class is not important but the order of calling them is very important.

LEC5 OOP 2018-2019

Accessing methods and instance variables

- Accessing state variables of an object: the . notation
- Using instance methods

Example:

```
public class Main {  
    public static void main(String[] args) {  
        Point p1=new Point();  
        p1.set(5,4);    p1.print();  
        Point p2;  
        p2=p1;  
        p2.set(7,8);    p2.print();    p1.print();    }  
    public class Point {  
        private int x,y;  
        public void set(int a , int b){ x=a; y=b;}  
        public void print(){    System.out.print(x+" ");    System.out.println(y); }}
```

The output is as shown:

```
5 4  
7 8  
7 8  
why?
```

Array of objects in Java

Returning to Point problem we can create ten points as shown in the following Processing program (using array):

```
public class Point {  
    private int x,y;  
    public void set(int a , int b){ x=a; y=b;}  
    public void print(){    System.out.print(x+" ");  
        System.out.println(y); }}
```

LEC5 OOP 2018-2019

```
Public class Main {  
Public static void main(String[] args){  
Point[] p=new Point[10];  
for (int i=0; i<p.length; i=i+1) {  
p[i] = new Point();  
p[i].set(L,i*2);  
p[i].print(); }  
}  
}
```

// we can solve it using three for loops one for creation, one for setting and one for printing.

H.W.

Define a class called Rectangle that has the following members:

Two integer attributes (variable members, attribute members) and two methods (method members) , area which returns the rectangle area while set_data is used to set the width and height.

Use this class to print:

1- one rectangle area 2- two rectangle areas using two object 3- three rectangle areas using one object 4- 10 rectangles areas' (use array)

Solution:

LEC5 OOP 2018-2019

1- & 2-

```
public class Rectangle {
private int w,h;
public int area(){return (w*h);}
public void set(int a, int b){w=a;h=b;}
}

public class Main {
    public static void main(String[] args) {
        Rectangle r1=new Rectangle();
        // two objects
        //Rectangle r2=new Rectangle();
        r1.set(10,20);
        //r2.set(1,2);
        System.out.println(r1.area());
        // System.out.println(r2.area());    }}
}
```

3-

```
public class Main {
    public static void main(String[] args) {
        Rectangle r1=new Rectangle();
        for(int i=1;i<=3;i++){
            r1.set( i,i*3);
            System.out.println(r1.area()); }    }}
}
```

LEC5 OOP 2018-2019

4-

```
public class Main {  
    public static void main(String[] args) {  
        Rectangle[] r_array=new Rectangle[10];  
        //or Rectangle r_array[]=new Rectangle[10];  
        for(int i=0;i<10;i++){  
            r_array[i]=new Rectangle();  
            r_array[i].set(i+1,(i+1)*2);  
            System.out.println(r_array[i].area());    }    }  
  
public class Rectangle {  
    private int w,h;  
    public int area(){return (w*h);}  
    public void set(int a, int b){w=a;h=b;}  
}
```