
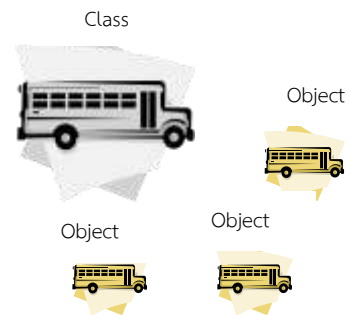


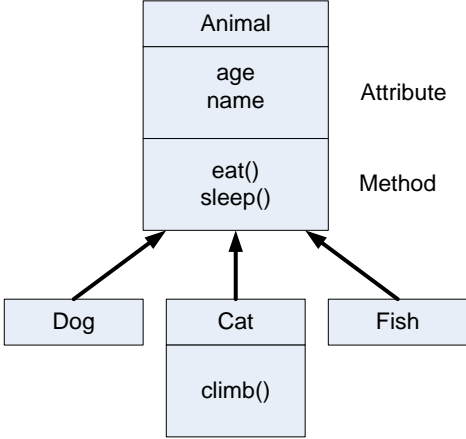
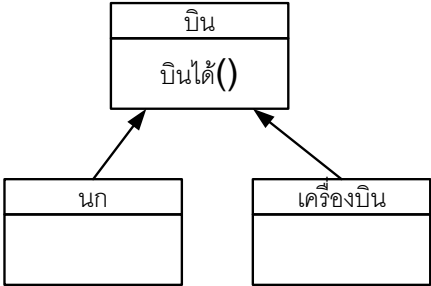


1.3 Object Class และคุณสมบัติของโปรแกรมเชิงวัตถุ

คุณสมบัติ	รายละเอียด
<p>Object</p> 	<p>วัตถุเป็นกุญแจสำคัญที่ทำให้เข้าใจเทคโนโลยีเชิงวัตถุ กล่าวคือ เมื่อมองไปรอบ ๆ จะพบตัวอย่างมากมายของวัตถุ เช่น สุนัข, โต๊ะ, โทรศัพท์, จักรยาน เป็นต้น</p> <p>วัตถุประกอบได้ด้วยคุณลักษณะ (attribute) และคุณสมบัติหรือการกระทำ (property or method)</p> <p>สุนัขมีคุณลักษณะ คือ ชื่อ, สี, พันธุ์, และการกระทำ คือ เห่า, กระดิกหาง</p> <p>คุณลักษณะและการกระทำของวัตถุเป็นวิธีคิดสำหรับการเขียนโปรแกรมเชิงวัตถุ</p>
<p>Class</p> 	<p>Class คือ นิยามหรือพิมพ์เขียวที่กำหนดไว้ล่วงหน้าก่อนจะเกิด objects หรือ instances โดยในพิมพ์เขียวจะประกอบไปด้วย กลุ่มของ attributes และ กลุ่มของ methods ที่แสดงค่าคุณลักษณะและการกระทำของ object</p>
<p>Abstraction</p> 	<p>การสร้างวัตถุแบบความคิดรวบยอด โดยการสร้างคลาสที่ประกอบไปด้วยคุณลักษณะ (attribute) และการกระทำ (method) ผู้ใช้ไม่จำเป็นต้องสนใจว่าคลาสนั้นมีรายละเอียดการทำงานอย่างไร ขอให้เรียกใช้งานได้อย่างถูกต้องผ่านทาง attribute และ method ก็พอ</p>

คุณสมบัติ	รายละเอียด
<p>Encapsulation</p> 	<p>การเขียนโปรแกรมลักษณะ OOP นั้น encapsulation ถือว่าเป็นองค์ประกอบหลักอย่างหนึ่ง ซึ่ง encapsulation หมายถึง การปกป้องและห่อหุ้ม attribute และ method เพื่อป้องกันข้อมูลหรือ method ของตนเองไม่ให้ถูกเข้าถึง โดยไม่ได้รับอนุญาต จาก object อื่นๆ</p> <p>ประโยชน์ของ encapsulation คือ ถ้าเราปล่อยให้ object อื่นเข้าถึงข้อมูลของเราได้อย่างอิสระข้อมูลที่เป็นความลับก็จะถูกเปิดเผยได้ หรือเมื่อมีการเปลี่ยนแปลงค่าข้อมูลก็จะทำให้ program ทำงานผิดพลาดได้ หรือเมื่อเราใช้ method อาจจะทำการเรียกใช้แบบผิดวิธี หรือผิดวัตถุประสงค์ได้</p>
<p>Inheritance</p> 	<p>คุณสมบัติการสืบทอด คือ คลาสลูกสามารถสืบทอด attributes และ methods จากคลาสแม่ได้ เปรียบเหมือน ลูกสาวสามารถสืบทอดรูปร่างหน้าตาและกิจกรรมมาจากแม่ตนเอง และคลาสลูกที่สืบทอดมานั้นยังสามารถพัฒนาต่อยอดได้ โดยเพิ่มเติมแก้ไข attributes หรือ methods ได้</p> <p>จากรูป class Animal เป็นคลาสแม่ ที่ประกอบไปด้วย attributes: age, name methods: eat(), sleep()</p> <p>Class Dog, Cat, Fish เป็นคลาสลูกที่สืบทอดมาจาก class Animal โดย class cat เพิ่ม method climb() เป็นต้น</p>
<p>Polymorphism</p> 	<p>การมีได้หลายรูปแบบ (many forms) เป็นคุณสมบัติที่ความสามารถของ method มีได้หลายรูปแบบขึ้นอยู่กับชนิดของวัตถุหรือพารามิเตอร์ที่ส่งไปทำงานของเมธอด เช่น คลาส bin มีเมธอด binได้() ซึ่งมีคลาส นก และเครื่องบิน เป็น subclass แต่ผลของการเรียกใช้เมธอด binได้() ของอ็อบเจกต์จากคลาสทั้งสองจะออกมาไม่เหมือนกัน</p>

ตัวอย่างการ นิยาม class และ การสร้าง object

<pre> class Employee { int age; String name; double salary; double id; double getSalary(){ return salary; } int getAge(){ return age; } } </pre>	<pre> Class TestEmployee { public static void main(String[] args){ Employee a = new Employee (); double vol; a.age = 10; a.salary = 20; vol = a.getSalary(); System.out.println(a.getAge()); } } </pre>
<pre> class Box { double width; double height; double depth; double vol(double w,double h,double d) { return w*h*d; } double vol() { return width*hight*depth; } } </pre>	<pre> Class TestBox { public static void main(String[] args){ Box a = new Box(); double vol; a.width = 10; a.height = 20; a.depth = 15; vol = a.width*a.height*a.depth; System.out.println("vol:" + vol); } } </pre>

จหาความแตกต่างระหว่าง code ทั้ง 2

<pre> class TestInsMethod1 { static int min(int m, int n) { if (m < n) return m; else return n; } static int max(int m, int n) { if (m > n) return m; else return n; } static int gcd(int m, int n) { int min = min(m, n); int max = max(m, n); if (max % min == 0) return min; else return gcd(min, max % min); } public static void main(String[] args){ int number1 = 25; int number2 = 20; System.out.println(gcd(number1, number2)); } } </pre>	<pre> class TestInsMethod1 { int min(int m, int n) { if (m < n) return m; else return n; } int max(int m, int n) { if (m > n) return m; else return n; } int gcd(int m, int n) { int min = min(m, n); int max = max(m, n); if (max % min == 0) return min; else return gcd(min, max % min); } public static void main(String[] args){ int number1 = 25; int number2 = 20; TestInsMethod1 a = new TestInsMethod1(); System.out.println(a.gcd(number1, number2)); } } </pre>
--	--

LAB 4

1. จงหาผลลัพธ์

```
int main(){

    int a[10],i,n,m,c,l,u;

    printf("Enter the size of an array: ");
    scanf("%d",&n);

    printf("Enter the elements of the array: " );
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }

    printf("Enter the number to be search: ");
    scanf("%d",&m);

    l=0,u=n-1;
    c=binary(a,n,m,l,u);
    if(c==0)
        printf("Number is not found.");
    else
        printf("Number is found.");

    return 0;
}

int binary(int a[],int n,int m,int l,int u){

    int mid,c=0;

    if(l<=u){
        mid=(l+u)/2;
        if(m==a[mid]){
            c=1;
        }
        else if(m<a[mid]){
            return binary(a,n,m,l,mid-1);
        }
        else
            return binary(a,n,m,mid+1,u);
    }
    else
        return c;
}
```

2. จงหาผลลัพธ์

```
void bubbleSort(int arr[], int n)
{
    if (n == 1)
        return;

    for (int i=0; i<n-1; i++)
        if (arr[i] > arr[i+1])
            swap(arr[i], arr[i+1]);

    bubbleSort(arr, n-1);
}

int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    for (int i=0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

3. จงหาผลลัพธ์

```
long memo[100] = {0};
long findValue(long n){

    if(memo[n] != 0)
        return memo[n];

    memo[n] = findValue(n-1)+ 10*findValue(n-1)+findValue(n-2);
    return memo[n];
}

int main(){
    long n;
    cin >> n;
    if(n > 50) return 0;
    memo[1] = 1;
    memo[2] = 1;
    cout << findValue(n) << endl;
    return 0;
}
```

4. Find Sum of Digits of a Number using Recursion

Enter the number: 2345
Sum of digits in 2345 is 14

5. Find the Biggest Number in an Array of Numbers using Recursion

Enter size of the list: 8

7 6 1 3 1 7 2 4

The largest number in the list is: 7

6. Recursion to Search an Element in Array Enter the size of the list:

10 3 6 7 5 3 5 6 2 9 1

Enter the key to search: 5

Key found at position: 6 Key found at position: 4

7. ค่าเฉลี่ย

กำหนดจำนวนเต็ม n จำนวน จงเขียนโปรแกรม หาค่าเฉลี่ยของข้อมูล n จำนวนดังกล่าว โดยให้เขียนอยู่ในรูป function recursive

Input

บรรทัดที่ 1 n แสดงจำนวนเต็ม n จำนวน $1 \leq n \leq 1000000$

บรรทัดที่ 2 i แสดงจำนวนเต็ม n จำนวนเว้นด้วยช่องว่าง $1 \leq i \leq 1000$

Output

บรรทัดที่ 1 ผลลัพธ์ค่าเฉลี่ยของข้อมูล n จำนวน จำนวนทศนิยม 2 ตำแหน่ง

ตัวอย่างผลลัพธ์

Input	Output
2 1 2	1.50
4 5 11 9 1	6.50
3 1 2 3	2.00