

Java의 클래스

[1] Java 클래스의 개요

- 클래스 = 필드(속성) + 메소드(함수)
- 객체 생성을 위한 설계도
- 아무리 작은 프로그램이라도 클래스를 만들어서 사용

[2] Java 문제 1

Java에서 클래스를 만들고 객체를 생성해서 사용하는 간단한 프로그램

```
class ClassA {  
    int a = 10;  
    int funcAdd(int x, int y) {  
        return x + y + a;  
    }  
}  
public class Test {  
    public static void main(String[] args) {  
        int x = 3, y = 6, r;  
        ClassA cal = new ClassA();  
        r = cal.funcAdd(x, y);  
        System.out.print(r);  
    }  
}
```

풀이

```
class ClassA { // 클래스를 정의  
    int a = 10;  
    int funcAdd(int x, int y) { // x=3, y=6  
        return x + y + a; // 입력받은값 3 + 6 + 10 = 19  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        int x = 3, y = 6, r;  
        ClassA cal = new ClassA(); // cal 객체생성  
        r = cal.funcAdd(x, y); // funcAdd(3, 6) 호출 = 19  
        System.out.print(r); // r = 19  
    }  
}  
결과 : 19
```

[3] Java 문제 2

Java 프로그램의 실행 결과를 확인

```
class ClassA {
    ClassA() {
        System.out.print('A');
        this.prn();
    }
    void prn() {
        System.out.print('B');
    }
}
class ClassB extends ClassA {
    ClassB() {
        super();
        System.out.print('D');
    }
    void prn() {
        System.out.print('E');
    }
    void prn(int x) {
        System.out.print(x);
    }
}
public class Test {
    public static void main(String[] args) {
        int x = 7;
        ClassB cal = new ClassB();
        cal.prn(x);
    }
}
```

풀이

```
class ClassA {
    ClassA() {
        System.out.print('A'); // (1) A 출력
        this.prn(); // 자식 메소드 prn()호출
    }
    void prn() {
        System.out.print('B'); //
    }
}

class ClassB extends ClassA {
    ClassB() {
        super(); // 부모(ClassA)호출
        System.out.print('D'); // (3) 'D' 출력
    }
}
```

```

    void prn() {
        System.out.print('E'); // (2) E 출력
    }
    void prn(int x) { // x = 7
        System.out.print(x); // 7 출력
    }
}

public class Test {
    public static void main(String[] args) {
        int x = 7;
        ClassB cal = new ClassB(); // cal객체 생성시 ClassB()생성자 실행
        cal.prn(x); // prn(7) 호출 => (4) 7 출력
    }
}
결과 : AED7

```

기출 따라잡기

[문제 1]

[문제 2]

```

public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length;
        for(int i = 0; i < b; i++)
            a[i] = i;
        return a;
    }
    public static void main(String[] args) {
        int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
    }
}

```

풀이

```

public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length; // b = 4
        for(int i = 0; i < b; i++) // i=0 1 2 3
            a[i] = i; // a[0]=0, a[1]=1, a[2]=2, a[3]=3
        return a; // [0, 1, 2, 3]
    }
    public static void main(String[] args) {

```

```

        int a[] = arr(); // a = [0, 1, 2, 3]
        for(int i = 0; i < a.length; i++) // i=0 1 2 3
            System.out.print(a[i] + " "); // 0 1 2 3
    }
}
결과 : 0 1 2 3

```

[문제 3]

```

class A {
    int a;
    public A(int a) { this.a = a; }
    void display() { System.out.println("a=" + a); }
}
class B extends A {
    public B(int a) {
        super(a);
        super.display();
    }
}
public class Test {
    public static void main(String[] args) {
        B obj = new B(10);
    }
}

```

풀이

```

class A {
    int a;
    public A(int a) { this.a = a; } // 10
    void display() { System.out.println("a=" + a); }
}

class B extends A {
    public B(int a) { // a = 10
        super(a); // super(10)
        super.display(); // a= 10
    }
}

public class Test {
    public static void main(String[] args) {
        B obj = new B(10); // obj 객체생성 매개변수 = 10
    }
}
결과 : a = 10

```

[문제 4]

```

class Test {
    public static void main(String args[]) {
        cond obj = new cond(3);
        obj.a = 5;
        int b = obj.func( );
        System.out.print(obj.a + b);
    }
}

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

```

풀이

```

class Test {
    public static void main(String args[]) {
        cond obj = new cond(3); // obj 객체 생성 , cond(int a)생성자호출
        obj.a = 5;
        int b = obj.func( ); // func() 함수 호출 b = 56
        System.out.print(obj.a + b); // 5 + 56 = 61
    }
}

class cond {
    int a; // a = 3, 5

    public cond(int a) { // a = 3
        this.a = a; // this.a = 3
    }

    public int func() { // func() 실행
        int b = 1;
        for (int i = 1; i < a; i++) // i = 1,2,3,4
            b += a * i; // b = b + a * i;
        return a + b; // 56
    }
}

```

결과 : 61

[문제 5]

```

class A {
    int a;
    int b;
}

public class Test {
    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);
    }
}

```

풀이

```

class A { // 클래스 A 생성
    int a; // 변수 a , 100
    int b; // 변수 b
}

public class Test {
    static void func1(A m) { // 공유 함수(메소드) func1(m)
        m.a *= 10; // m.a = m.a * 10;
        // m.a = 100 * 10 = 1000;
    }
    static void func2(A m) {
        m.a += m.b; // m.a = m.a + m.b
        // m.a = 1000 + 1000 = 2000
    }
    public static void main(String args[]) { // main()실행함수
        A m = new A(); // m객체 생성 A클래스
        m.a = 100; //
        func1(m); // 함수(메소드) 호출, m.a = 1000
        m.b = m.a; // m.b = 1000
        func2(m); // 함수(메소드) 호출, m.a = 2000
        System.out.printf("%d", m.a); // 2000
    }
}

```

```
}  
결과 : 2000
```

[문제 6]

```
class Parent {  
    int x = 1000;  
    Parent() {  
        this(3000);  
    }  
    Parent(int x) {  
        this.x = x;  
    }  
}  
  
class Child extends Parent {  
    int x = 4000;  
    Child() {  
        this(5000);  
    }  
    Child(int x) {  
        this.x = x;  
    }  
    int getX() {  
        return this.x;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        Child c = new Child();  
        System.out.println(c.getX());  
    }  
}
```

풀이

```
class Parent {  
    int x = 1000;  
    Parent() {  
        this(3000);  
    }  
    Parent(int x) {  
        this.x = x;  
    }  
}  
  
class Child extends Parent { // Child 클래스가 Parent 클래스를 상속받음  
    int x = 4000; // x = 5000
```

```
Child() {  
    this(5000); // Child(int x) 호출 Child(5000)  
}  
Child(int x) { // x = 5000  
    this.x = x; // this.x = 5000  
}  
int getX() {  
    return this.x; // this.x = 5000  
}  
}  
  
public class Test {  
    public static void main(String[] args) {  
        Child c = new Child(); // Child 클래스를 사용하여 c라는 객체 생성  
        System.out.println(c.getX());  
    }  
}
```

- **this**: 객체 자신을 가리키는 레퍼런스 변수로, 자신의 객체에 접근할 때 사용됩니다
- **this()**: 같은 클래스에서 생성자가 다른 생성자를 호출할 때 사용됩니다

결과 : 5000