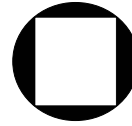


Repaso/resumen

1. lectura/escritura y asignación
2. funciones
3. selección de instrucciones (if-else)
4. repetición de instrucciones (while)
5. recursión
6. operadores/instrucciones especiales

Lectura/escritura y asignación

Problema. Calcular el área y perímetro de la figura:



```
U.println("Círculo y cuadrado inscrito");
double r=U.readDouble("radio?");
double a=2*r/1.41;//lado cuadrado
U.println("perímetro=" + (2*Math.PI*r+4*a));
U.println("área=" + (Math.PI*r*r-a*a));
```

Nota. Programa lineal o secuencial (instrucciones se ejecutan en orden de aparición)

?: operador de resto de división

//invertir entero de 3 dígitos

```
int n=U.readInt("Nº 3 dígitos?");
int d1=n/100, d3=n%10;
int d2=n%100/10;//int d2=n/10%10;
U.println("inverso="+d3+d2+d1);
//cajero automático
int d=U.readInt("¿Cuánto dinero necesita?");
U.println(d/20000+" de $20.000");d=d%20000;
U.println(d/10000+" de $10.000");d=d%10000;
U.println(d/5000+" de $5.000");d=d%5000;
U.println(d/2000+" de $2.000");d=d%2000;
U.println(d/1000+" de $1.000");
```

Programas lineales con ejecución de funciones predefinidas

```
U.println("triángulo: lados a,b,y angulo alfa");
//obtener datos
double a=U.readDouble("a ? "),
b=U.readDouble("b ? "),
alfa=U.readDouble("alfa ? ") *Math.PI/180;

//calcular tercer lado
double c=Math.sqrt(Math.pow(a*Math.sin(alfa),2)+
Math.pow(b-a*Math.cos(alfa),2));

//calcular y mostrar perimetro y area
U.println("Perímetro=" + (a+b+c));
double s=(a+b+c)/2; //semi-perímetro
U.println("Area="+Math.sqrt(s*(s-a)*(s-b)*(s-c)));
```

Programas lineales con ejecución de funciones definidas por el programador

```
//seleccionar mayor entre dos fechas
static public void main(String[]args)throws IOException{
    int f1=U.readInt("fecha1(ddmmaaaa)?"),
    f2=U.readInt("fecha2(ddmmaaaa)?");
    U.println("mayor="+
        ddmmaaaa(Math.max(aaaammdd(f1),aaaammdd(f2))));
}
//convertir fecha ddmmaaaa a la forma aaaammdd
static public int aaaammdd(int x){
    int dd=x/1000000, mm=x/10000%100, aaaa=x%10000;
    return aaaa*10000 + mm*100 + dd;
}
//convertir fecha aaaammdd a la forma ddmmaaaa
static public int ddmmaaaa(int x){
    int dd=x%100, mm=x%10000/100, aaaa=x/10000;
    return dd*1000000 + mm*10000 + aaaa;
}
```

Selección instrucciones (if-else)

```
static public int diasMes(int m,int a){
    if(m==1||m==3||m==5||m==7||m==8||m==10||m==12)
        return 31;
    else if(m==4||m==6||m==9||m==11)
        return 30;
    else if(m==2)//if(cond simple) sin else
        if(bisiesto(a))
            return 29;
        else
            return 28;
    return 0;//mes incorrecto
}
static public boolean bisiesto(int x){
    return x%4==0 && x%100!=0 || x%400==0;
}
```

Clase 7: Repaso/Resumen

Repetición instrucciones (while)

```
//tabla celsius - fahrenheit
int g=-20;
while(g<=40){
    U.println(g+"°C="+ (9.0/5.0*g+32)+"°F");
    g=g+5;
}
static public double potencia (double x,int y){
    if(y==0 && x==0) U.abortar("0^0 indefinido");
    double producto=1;
    int i=2;
    while(i<=Math.abs(y)){
        producto=producto*x;
        i=i+1;
    }
    if(y>=0)
        return producto;
    else
        return 1/producto;
}
```

Patrones para proceso de listas

<pre>//inicializar int mayor=0; //leer 1º int n=U.readInt("n°?"); //repetir mientras while(n!=0){ //procesar if(n>mayor) mayor=n; //leer sgte n=U.readInt("n°?"); } //finalizar U.println("mayor="+mayor);</pre>	<pre>//inicializar int mayor=0; //repetir siempre while(true){ //leer int n=U.readInt("n°?"); //condición de término if(n==0) break; //procesar mayor=Math.max(mayor,n); } //finalizar U.println("mayor="+mayor);</pre>
---	---

```
static public int factorial(int x){
    int producto=1;
    int i=1
    while(i<=x){
        producto=producto*i;
        i=i+1;
    }
    return producto;
}
static public int factorial(int x){
    int producto=1;
    while(x>0){
        producto=producto*x;
        x=x-1; //no modifica arg en llamada
    }
    return producto;
}
```

```
//combinaciones(x,y)=x!/(y!(x-y)!)
static public int combinaciones(int x,int y){
    return factorial(x)/(factorial(y)*factorial(x-y));
}
Nota. No sirve para números mayores que 12

//combinaciones(x,y)=((x-y+1)*...*x)/(1*...*y)
static public int combinaciones(int x,int y){
    //p=(x-y+1)*...*x
    int p=1, i=x-y+1;
    while(i<=x){p=p*i; i=i+1;}
    //p/(2*...*y)
    int j=2;
    while(j<=y){p=p/j; j=j+1;}
    return p;
}
```

Funciones Recursivas

```
//factorial(x)
static public int factorial(int x){
    if(x==0)
        return 1; //caso base
    else
        return x*factorial(x-1);
}
static public int factorial(int x){
    if(x==0) return 1; //caso base
    return x*factorial(x-1);
}
```

Funciones Recursivas

```
//fibonacci(i)=i-ésimo de 0,1,1,2,3,5,8,13,21,...
static public int fibonacci(int i){
    if(i<=2) return i-1;
    return fibonacci(i-1)+fibonacci(i-2);
}
//combinaciones(x,y)=x!/(y!(x-y)!)
//c(x,y)=c(x-1,y)+c(x-1,y-1); c(x,x)=c(x,0)=1
static public int combinaciones(int x,int y){
    if(x==y || y==0) return 1;
    return combinaciones(x-1,y)
        + combinaciones(x-1,y-1);
}
Nota. Combinaciones(10,2)=45
Ej: combinaciones de 10 esferas numeradas 0 a 9 tomados de a 2
```

Clase 7: Repaso/Resumen

Métodos void recursivos

```
static public void misterio(){
    int n=U.readInt("n°?");
    if(n==0) return;
    misterio();
    U.println(n);
}
¿Qué hace misterio(); con los sgtes datos?
n°? 6
n°? 3
n°? 5
n°? 0
```

Operadores especiales

```
double suma, mayor, numero; int n;
suma = mayor = n = 0;
while((numero=U.readDouble("n°?"))>0){
    U.println("prom="+(suma+=numero)/++n);
    mayor=max(numero,mayor);
}
U.println("mayor="+mayor);

static public double max(double x,double y){
    return x>y ? x : y;
}
```

? en Funciones Recursivas

```
//fibonacci(i)=i-ésimo de 0,1,1,2,3,5,8,13,21,...
static public int fibonacci(int i){
    return i<=2? i-1 :
           fibonacci(i-1)+fibonacci(i-2);
}
//combinaciones(x,y)=x!/(y!(x-y)!)
static public int combinaciones(int x,int y){
    return x==y || y==0 ? 1:
           combinaciones(x-1,y)
           + combinaciones(x-1,y-1);
}
```

Instrucciones especiales

```
static public void combinacionesDigitos(){
    for(int x=0,n=0; x<=9; ++x)
        for(int y=x+1; y<=9; ++y)//int y=0?
            if(x!=y) U.println(++n +": "+x +", "+y);
}
static public int diasMes(int m,int a){
    switch(m){
    case 1:case 3:case 5:case 7:case 8:case 10:case 12:
        return 31;
    case 4:case 6:case 9:case 11:
        return 30;
    case 2: return bisiesto(a) ? 29 : 28;
    }
    return 0;
}
```

```
static public void printNumero2Digitos(int x){
    switch(x){
    case 11: U.print("once"); return;
    case 12: U.print("doce"); return;
    case 13: U.print("trece"); return;
    case 14: U.print("catorce"); return;
    case 15: U.print("quince"); return;
    default:
        int d1=x/10, d2=x%10;//primer y 2° dígito
        switch(d1){
        case 1: U.print("diez"); break;
        case 2: U.print("veinte"); break;
        case 3: U.print("treinta"); break;
        case 4: U.print("cuarenta"); break;
        case 5: U.print("cincuenta");break;
        case 6: U.print("sesenta"); break;
        case 7: U.print("setenta"); break;
        case 8: U.print("ochenta"); break;
        case 9: U.print("noventa"); break;
        }
        if(d2==0 && d1!=0) return;//10,20,...,90
        if(d1>0) U.print(" y "); //1x,2x,...,9x
        printDigito(d2);
    }
}
```

```
static public void digitoRomano(int x){
    switch(x){
    case 3: I();
    case 2: I();
    case 1: I(); break;
    case 4: I();
    case 5: U.print("V"); break;
    case 6: case 7: case 8:
        U.print("VI");
        switch(x){case 8: I(); case 7: I();}
        break;
    case 9: U.print("IX");break;
    }
}
static public void I(){
    U.print("I");
}
```