

Clase1: lectura/escritura/asignación

Problema.

Escribir un programa que calcule los porcentajes de hombres y mujeres de acuerdo al siguiente diálogo con una persona (usuario)

nº de hombres? 521 ← persona

nº de mujeres? 140 ← persona

% de hombres = 78.8

% de mujeres = 21.2

Algoritmo. Etapas (pasos, instrucciones) que debe realizar (ejecutar) el computador para resolver el problema

1. **escribir** (mostrar) en la pantalla la frase "nº de hombres?"
2. **leer** (obtener) el nº ingresado por la persona usando el teclado
3. **escribir** (mostrar) en la pantalla "nº de mujeres?"
4. **leer** el nº ingresado por la persona
5. **calcular** el % (p) de hombres c/r a suma de hombres y mujeres
6. **escribir una línea con la frase** "% de hombres =" y el nº p
7. **escribir una línea con** "% de mujeres =" y el nº 100-p

Programa. Algoritmo expresado en un lenguaje de programación

Programa en lenguaje Java (instrucciones principales):

```
U.print("nº de hombres?");  
  
int h = U.readInt();  
  
U.print("nº de mujeres?");  
  
int m = U.readInt();  
  
double p = 100.0*h/(h+m);  
  
U.print("% de hombres = "); U.println(p);  
  
U.print("% de mujeres = "); U.println(100-p);
```

Lenguaje Java: características generales

- inspirado en idioma inglés
 - palabras int, print, read,...
- uso de notación matemática
 - expresiones: $100.0 * h / (h + m)$, $p - 100$
 - funciones: print(...), readInt()
- sintaxis (gramática, escritura) reglamentada
 - instrucciones terminan con ;
 - palabras con significados especiales (int, double, print)
 - uso de minúsculas (salvo casos especiales)
- semántica (interpretación) precisa (sin ambigüedades)
 - instrucciones con significado/efecto bien definido

Explicaciones detalladas

```
U.print("nº de hombres ? ");
```

escribe en la pantalla la frase "nº de hombres ? "

Sintaxis: `U.print("...");`

Semántica:

Función (método) que escribe parámetro (argumento) en la pantalla del computador

```
int h = U.readInt();
```

lee un número entero desde el teclado y lo guarda en la variable h

Equivalencia

```
int h; //declaración
```

```
h = U.readInt(); //asignación
```

Nota.

```
//frase
```

es un comentario que contiene observaciones de utilidad para el programador, pero que es ignorado por el computador al ejecutar las instrucciones del programa

Clase1: lectura/escritura/asignación

```
int h;
```

Semántica: declara (define) h como una variable entera

¿variable?

- representación simbólica de un valor (número)
- representa una ubicación (celda) en la memoria del computador
- capacidad: un valor/número (del tipo indicado)
- puede variar (cambiar) su valor

Sintaxis: tipo nombre;

- int: n° entero de hasta 9 dígitos. Ejs: 123, -4, 2147483647 (mayor)
- double: n° real de hasta 15 dígitos de precisión. Ejs: 100.0, 0.001
- nombre (identificador): letra seguida de letras, dígitos o_

```
h = U.readInt();
```

Semántica:

1° método (función) U.readInt()

1. espera que el usuario ingrese un número (dígitos y enter)
2. lee (obtiene) el número
3. entrega el número como resultado

2° número se guarda en (asigna a) la variable h

Sintaxis: variable = U.readTipo();

readTipo(): readInt() o readDouble()

```
p=100.0*h/(h+m);
```

calcula $100.0 * h / (h + m)$ y guarda el resultado en la variable p

Instrucción de asignación

sintaxis: variable=expresión;

semántica

- 1° evaluar expresión (calcular resultado)
- 2° asignar (guardar) resultado a (en) la variable

¿expresión?

- se escribe en una línea (hacia el lado) y no en varios niveles
- componentes:
 - variables y constantes (enteras y reales)
 - operadores binarios (+, -, *, /) y unarios (+, -)
 - métodos (funciones) que entregan un número (Ej:readInt)
 - expresiones con o sin paréntesis

Evaluación de expresión

- 1° operadores unarios (+,-)
- 2° operadores “multiplicativos” (*, /)
- 3° operadores “aditivos” (+, -)

Notas

- en caso de operadores de igual prioridad, evaluar de izq a der.
Ej: $100.0 * h / (h + m)$ equivale a $(100.0 * h) / (h + m)$
- paréntesis se usan para
 - modificar orden de evaluación. Ej: $100.0 * h / (h + m)$
 - confirmar orden de evaluación. Ej: $(100.0 * h) / (h + m)$

Tipo del resultado

- tipo común. Ej: 1 / 2 es 0 (entero), 1.0 / 2.0 es 0.5 (real)
 - real si operandos son de tipos distintos. Ej: 1.0/2 es 0.5
- Nota. ¿resultado de $h / (h + m) * 100.0$? ¿por qué?

```
U.println(p);
```

Semántica: escribe el valor de la variable p y “salta” a la sgte línea

- 1° evaluar expresión (ejs: p, 100-p)
- 2° escribir resultado
- 3° posicionar el cursor al comienzo de la sgte línea en la pantalla

Sintaxis: U.print(ln)(expresión aritmética);

expresión aritmética: operando operador operando

operador: +, -, *, /

operando:

- variable (ej: p)
- constante (ej: 100)
- expresión con/sin paréntesis (ejs: a*b, (a/b))

Solución 2. Usando abreviaciones

```
int h = U.readInt("n° de hombres?");
int m = U.readInt("n° de mujeres?");
double p = 100.0 * h / (h+m);
U.println("% de hombres = " + p);
U.println("% de mujeres = " + (100-p));
```

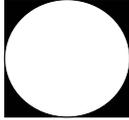
Equivalencias:

```
int h = U.readInt("n° de hombres?");
es equivalente a
U.print("n° de hombres?"); int h=U.readInt();
```

```
U.println("% de hombres = " + p);
es equivalente a
U.print("% de hombres = "); U.println(p);
```

Clase1: lectura/escritura/asignación

Problema. Para calcular el área y perímetro de la figura formada por un cuadrado y un círculo inscrito



escriba un programa que establezca el siguiente diálogo

```
lado cuadrado? ... (nº real)
área=nº
perímetro=nº
```

Solución 1 (sin abreviaciones)

```
U.print("lado cuadrado?");
double x = U.readDouble();
double r = x/2;
U.print("área=");
U.println(x*x - 3.1416*r*r);
U.print("perímetro=");
U.println(4*x + 2*3.1416*r);
```

Solución 2 (con abreviaciones)

```
double x = U.readDouble("lado cuadrado?");
double r = x/2;
double pi = 3.1416;
U.println("área=" + (x*x - pi*r*r));
U.println("perímetro=" + (4*x + 2*pi*r));
```

Solución 3 (con algunas novedades útiles)

```
double x=U.readDouble("lado cuadrado?"),r=x/2;
U.println("área=" +
(Math.pow(x,2) - Math.PI*Math.pow(r,2));
U.println("perímetro=" + (4*x + 2*Math.PI*r));
```

double x=...,r=...;

- equivalencia: double x=...; double r=...;

Math.PI

- valor predefinido de π (3.14159265358979)
- constante real (valor inmodificable)

Math.pow(r, 2)

- calcula y entrega r^2
- Math.pow: función predefinida (incorporada al lenguaje Java)
- sintaxis: Math.pow(x, y), x e y: expresiones
- semántica: x^y , con resultado real

Programa completo

```
import java.io.*;
//todo programa debe estar contenido
//en una clase class Nombre{...}
//y grabarse en archivo Nombre.java
class Programa
{
//encabezamiento del método principal (main)
static public void main(String[]args)
throws IOException
{
... //instrucciones
}
}
Notas
```

- burocracia necesaria porque Java no se diseñó para principiantes
- lectura y escritura es complicada, pero se simplifica usando la clase U (clase de la U con funciones para leer y escribir frases y números)

```
import java.io.*;
class U{ //en archivo U.java
//declaración de entrada estandar (teclado)
static public BufferedReader teclado =
new BufferedReader(new InputStreamReader(System.in));
//funciones (métodos) para leer enteros y reales
static public String readLine(String x)throws IOException{
print(x); return teclado.readLine();
}
static public int readInt(String x)throws IOException{
return Integer.parseInt(readLine(x));
}
static public int readInt()throws IOException{
return readInt("");
}
static public double readDouble(String x)throws IOException{
return Double.parseDouble(readLine(x));
}
static public double readDouble()throws IOException{
return readDouble("");
}
}
```

```
//métodos para escribir frases, enteros y reales
static public void print(String x){
System.out.print(x);
}
static public void print(int x){
System.out.print(x);
}
static public void print(double x){
System.out.print(x);
}
static public void println(String x){
System.out.println(x);
}
static public void println(int x){
System.out.println(x);
}
static public void println(double x){
System.out.println(x);
}
}
} //fin de clase U
```