



Universidad Nacional de San Luis
Área de Programación y Metodologías de Desarrollo de Software
Ingeniería en Informática – Ingeniería en Computación
Programación II
Práctico N° 1
Lenguaje Java: Tipos, Variables, Expresiones y Sentencias

Ejercicio 1: defina los siguientes conceptos:

- a) Tipo Primitivo
- b) Tipo Referenciado.

Ejercicio 2: ¿Por qué es importante tener muy en claro, a la hora de programar, los conceptos mencionados en el ejercicio anterior? Justifique su respuesta.

Ejercicio 3: dado el siguiente programa:

```
public class Precedencia {  
    public static void main(String[] args) {  
        int x = 1, y = 2, z = 3;  
        int a = x + y - 2/2 + z;  
        int b = x + (y - 2)/(2 + z);  
        System.out.println("a = " + a + " b = " + b);  
    }  
}
```

Se pide:

- a) Mostrar cual es la salida por pantalla.
- b) Ordenar, por cada sentencia, los operadores utilizados según su precedencia.

Ejercicio 4: realiza un programa que permita saludar al usuario de acuerdo al nombre ingresado.

Ejercicio 5: declara dos variables numéricas, muestra por consola la suma, resta, multiplicación, división de los valores asignados a las mismas.

Ejercicio 6: desarrolla un programa que pida por teclado 5 valores y los muestre en orden inverso y duplicado su valor. Es decir, si el ingreso fue 2, 3, 5, 1, 3 el resultado por pantalla será 6, 2, 10, 6, 4.

Ejercicio 7: realiza un programa que permita cargar al usuario un número x entre 0 y 10. El programa debe mostrar por pantalla la tabla de multiplicar (del 1 al 10) del número ingresado.

Ejercicio 8: realiza una aplicación que permita al usuario ingresar número impares, los almacene y los imprima por pantalla. El programa finaliza cuando el usuario ingresa un número par.

Ejercicio 9: escribe un programa que permita el ingreso de letras del alfabeto y cuente las letras repetidas. El programa se detiene cuando algunas de las letras se repitieron un número k de veces. El número k es ingresado por el usuario.



Por ejemplo: si se ingresa el valor 3 para k y la secuencia “a”, “r”, “e”, “u”, “a”, “r”, “l” y “a”; el programa debe parar de solicitar caracteres luego de la última “a” (se repitió 3 veces) y debe mostrar lo siguiente: a=3, r=2, e=1, u=1, l=1.

Nota: use el tipo de dato **String** para el ingreso de cada letra.

Ejercicio 10: declara un arreglo de tamaño TAM (constante). Si TAM es impar entonces se deben cargar los primeros “TAM” valores impares y si TAM es par se deben cargar los primeros “TAM” valores pares en dicho arreglo. Finalmente muestra los valores por pantalla.

Por ejemplo: si TAM es 3, debería ingresar en el arreglo los valores 1, 3 y 5.

Ejercicio 11: explique y de ejemplos de cómo funcionan las sentencias **break** y **continue**.

Ejercicio 12: Dado el siguiente programa:

```
import java.io.*;
class Bucles {
    public static void main (String argv[ ]) {
        int i=0;
        for (i=1; i<5; i++) {
            System.out.println("antes "+i);
            if (i==2) continue;
            if (i==3) break;
            System.out.println("después "+i);
        }
    }
}
```

Se pide:

- Ejecuta el programa mostrando claramente la salida y el flujo de ejecución.
- Modifica el programa para que la iteración tenga un único punto de entrada y un único punto de salida.

Ejercicio 13: desarrolla un programa que cuente la cantidad de vocales en una cadena ingresada por el usuario.

Nota: recorre el string con el método charAt(x). Este método retorna el carácter que está en la posición “x” del string. Por ejemplo, si String s= “Hola”, entonces s.charAt(2) devuelve el carácter “l”.

Ejercicio 14: dado el siguiente programa:

```
class BreakDemo {
    public static void main(String[] args) {
        int[] arrayOfInts = {32, 87, 3, 589, 12, 1076, 2000, 8, 622, 127};
        int searchfor = 12;
        int i;
        boolean foundIt = false;
        for (i = 0; i < arrayOfInts.length; i++) {
            if (arrayOfInts[i] == searchfor) {
                foundIt = true;
                break;
            }
        }
    }
}
```



```
}  
}  
if (foundIt) {  
    System.out.println("Found " + searchfor + " at index " + i);  
} else {  
    System.out.println(searchfor + " not in the array");  
}  
}  
}
```

Se pide:

- Ejecuta el programa mostrando claramente la salida y el flujo de ejecución.
- Modifica la iteración para que la misma tenga un único punto de entrada y un único punto de salida.

Ejercicio 15: desarrolla un programa que mediante un menú admita reservar o cancelar asientos de un avión, así como mostrar qué asientos están ocupados y libres actualmente. El avión posee 25 filas de asientos y 4 columnas.

Ejercicio 16: dado el siguiente programa:

```
import java.util.*;  
public class ForEachFloat {  
    public static void main(String[] args) {  
        Random rand = new Random(47);  
        float f[] = new float[10];  
        for(int i = 0; i < 10; i++)  
            f[i] = rand.nextFloat();  
        for(float x : f)  
            System.out.println(x);  
    }  
}
```

Se pide:

- Diga que hace la sentencia: **for (float x : f)**
- Ejecute el programa.

Ejercicio 17: desarrolla un programa que determina la cantidad de dinero que cobrará un trabajador en el mes. Para esto se sabe que cuando las horas de trabajo mensuales exceden de 40, el resto se contabiliza como horas extras y que estas se pagan al doble de una hora normal cuando no exceden de 8; si las horas extras exceden de 8 se pagan las primeras al doble de lo que se paga por una hora normal y el resto el triple.

Nota: el programa debe solicitar el valor en pesos que vale una hora normal, la cantidad total de horas trabajadas y mostrar por pantalla el monto total que va a cobrar el trabajador.



EJERCICIOS COMPLEMENTARIOS

Ejercicio a: explica la diferencia entre conversiones forzadas por el programador (cast) y las conversiones automáticas.

Ejercicio b: elabora un programa que permita al usuario ingresar tres números enteros y determine cuál es el menor de ellos.

Nota: Para la solución de este ejercicio no debes utilizar iteraciones.

Ejercicio c: realiza un programa que lea un número por teclado y muestre por consola el carácter al que pertenece en la tabla ASCII. Por ejemplo: si introduzco un 97, me muestre una “a”.

Ejercicio d: lee un número por teclado que pida el precio de un producto (puede tener decimales) y calcule el precio final con IVA. El IVA será una constante que será del 21%.

Ejercicio e: construye un programa que encuentre el menor de una secuencia de números ingresada por el usuario. El programa debe permitir el ingreso de números hasta que el usuario ingrese un -1.

Ejercicio f: declara un arreglo de tipo float en el cual se carguen los primeros 20 números impares automáticamente. Para esto use la sentencia for.

Ejercicio g: construye un programa que permita al usuario ingresar la cantidad y los valores de calificaciones y calcule el promedio total.

Ejercicio h: desarrolla un programa que ayude a un cajero a identificar la cantidad de billetes y monedas que se necesitan de cada una de las siguientes denominaciones: \$100, \$50, \$20, \$10, \$5, \$2, \$1 y \$0.50 para una cantidad determinada. Siempre debe tener como principal criterio usar los billetes de mayor denominación. Por ejemplo, si la cantidad es de \$392,50, se necesitan 3 billetes de \$100, 1 de \$50, 2 de \$20, 2 de \$2 y una moneda de \$0.50.

Ejercicio i: Realiza las siguientes tareas:

- a) Desarrolla un menú de opciones con las características que se muestran a continuación:

“MENU DE OPCIONES”

- 1) Crear un arreglo de tamaño x
 - 2) Ingresar sólo números pares en el arreglo
 - 3) Ingresar sólo números impares en el arreglo
 - 4) Calcular la tabla de multiplicar del número x
 - 5) Imprimir el arreglo cargado
 - 6) Ingresar el nombre del usuario y saludarlo
- Ingrese una opción:

- b) Implementa las acciones indicadas por cada opción.

Nota 1: el programa debe mostrar todos los carteles correspondientes. Por ejemplo, si el usuario quiere llenar el arreglo antes de crearlo, se le debe advertir que primero se debe crear la estructura.



Nota 2: en caso de que el usuario ingrese una opción que no figura debe indicar que lo ingrese nuevamente.

Ejercicio j: teniendo en cuenta el ejercicio de los billetes (ejercicio h), implementa el mismo programa del cajero considerando una capacidad limitada de billetes. Es decir, el cajero debe considerar la cantidad de billetes que posee para poder expender las sumas solicitadas.

Ejercicio k: escriba un programa que declare una variable *num* de tipo entero y que permita asignarle un valor por teclado. A continuación, muestre un mensaje indicando si el valor de *num* es positivo o negativo. Para este ejercicio particular se considera el 0 como positivo. Para resolver el ejercicio debe utilizar el operador condicional (`? :`) dentro de un `println()`.

Ejercicio l: realice un programa que lea 10 números enteros por teclado y los guarde en un array. Calcule y muestre la media de los números que estén en las posiciones pares del array. Considere el caso particular de la primera posición del array (posición 0).