



UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732



Acreditación Institucional
Internacional

OTORGADA POR EL IAC CINDE ACUERDO 55 DEL 8 DE MAYO-VIGENCIA 5 AÑOS



ACREDITACIÓN
INSTITUCIONAL
DE ALTA CALIDAD
MULTICAMPUS

Res. MEN No. 03495 del 29 de enero de 2010

Vigencia por seis años



QS STARS
RATED FOR EXCELLENCE



ISO 9001
Acreditación

ISO 9001-1



CERTIFIED
IONet
MANAGEMENT SYSTEM



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

Faculty: Systems engineer
Course: Introduction of Programming
Topic: Ciclos FOR y WHILE

Socializer: Luis Fernando Castellanos Guarín
Karen Daniela Cuervo Cely

Email: Luis.castellanosg@usantoto.edu.co
Karen.cuervo@usantoto.edu.co

Phone: 3214582098
3105856930



Topics

Cycles “FOR” and “WHILE”

- About / Acerca de
- Como funciona /How it works
- Sintaxis / Syntax
- Ejemplos / examples
- Ejercicios / Exercises

¡Siempre
hacia lo alto!





About cycles

The **cycles**, or known as loops, are high-importance control structures in most software. They are used in JAVA and virtually all programming languages used today (2020). The loops allow us to nometer one or more instructions (lines of code how much) times we take it or necessary sea.

A helps us carry out a repetitive task on a very small number of lines without virtually form.

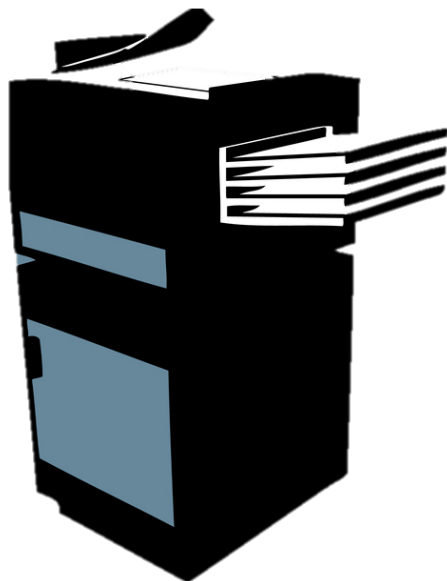
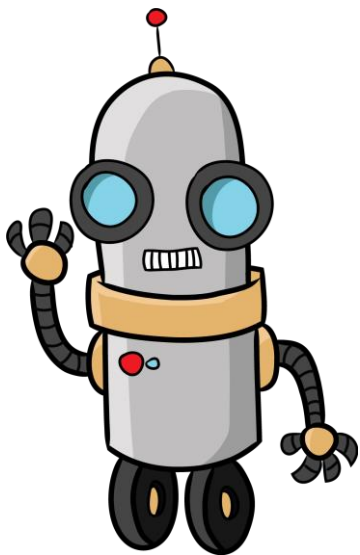


Los **ciclos** o también conocidos como **bucles**, son estructuras de control de alta importancia en la mayoría de los software. Se usan en JAVA y prácticamente en todos los lenguajes de programación usados en la actualidad (2020). Los bucles nos permite repetir una o varias instrucciones (líneas de código) cuantas veces lo necesitemos o sea necesario.

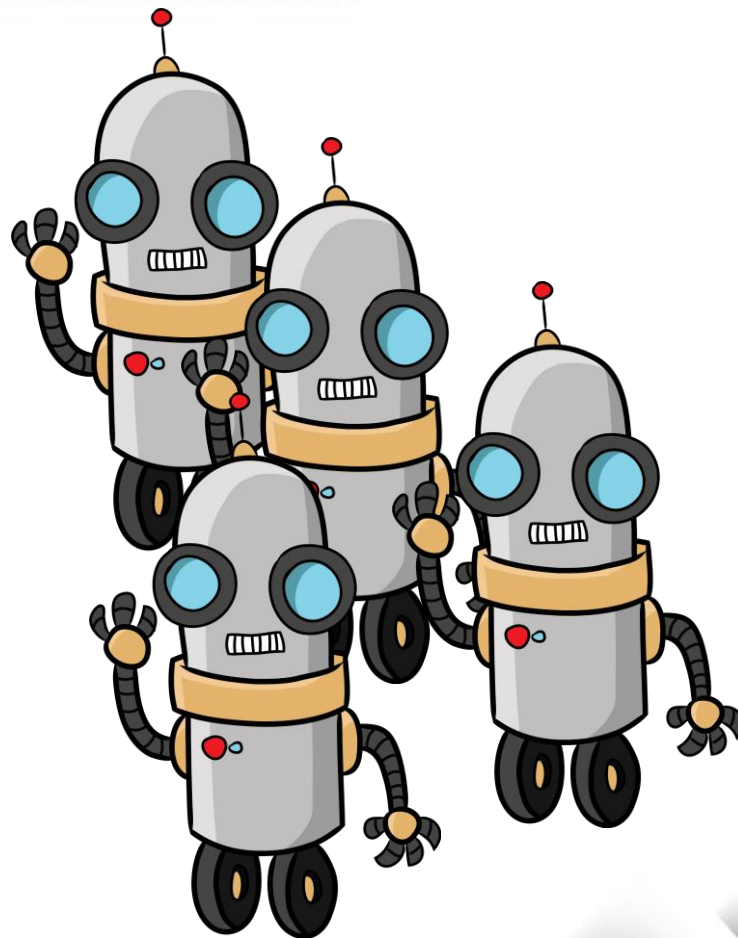
Un ciclo nos ayuda a llevar a cabo una tarea repetitiva en una cantidad de líneas muy pequeña y de forma prácticamente automática.



How it works



Repetidora de código
FOR
WHILE

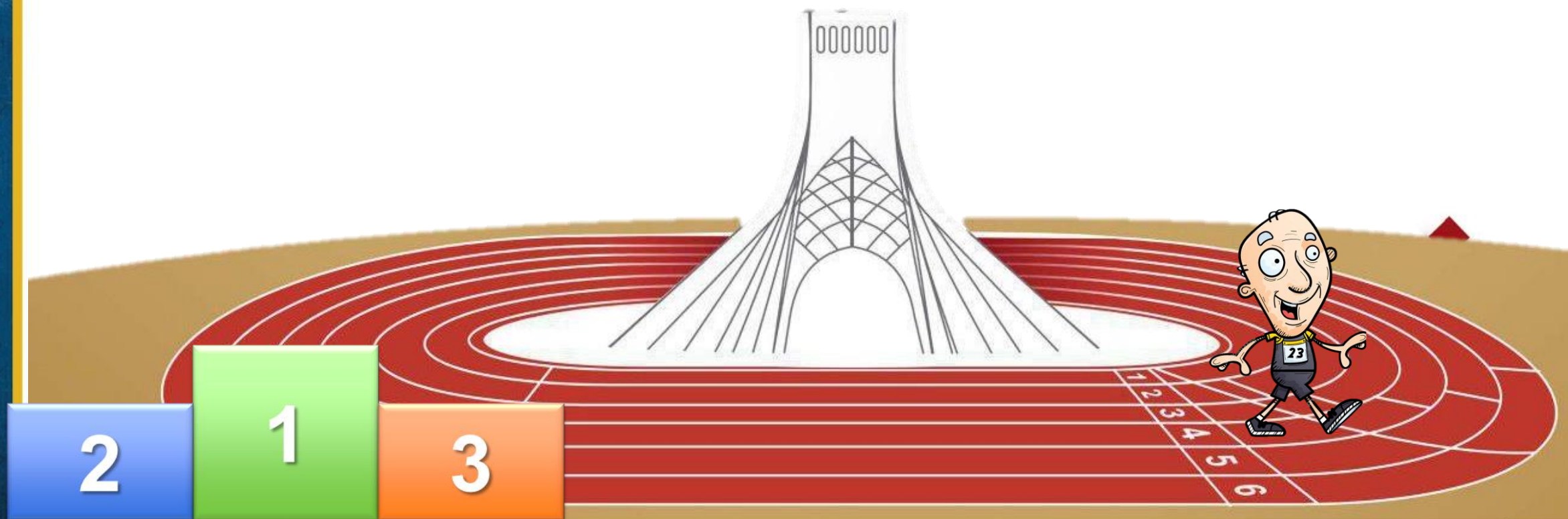


¡Siempre
hacia lo alto!



Cycles “for”

*“Hacer lo mismo **N** veces”*





Syntax

La sintaxis de un ciclo FOR es simple en lenguaje JAVA, en realidad en la mayoría de los lenguajes de alto nivel es incluso muy similar, de hecho, con tan solo tener bien claros los 3 componentes del ciclo for (inicio, final y tamaño de paso) tenemos prácticamente todo hecho:

```
1  for(int i = valor inicial; i <= valor final; i = i + paso)
2  {
3      ....
4      ....
5      Bloque de Instrucciones....
6      ....
7      ....
8  }
```

- **Línea 1:** Creamos una variable de control llamada **i** que es tipo entero (int), se le asigna un valor inicial y un valor final con el cual determinamos cuantas veces se va a repetir el ciclo.
- **Línea 2:** "{" con la cual damos por iniciado el ciclo.
- **Línea 3 a 7:** En estas líneas es donde estarán todo el código fuente que queramos llevar a cabo de manera iterativa durante la ejecución del ciclo
- **Línea 8:** "}" con la cual damos por cerrado el ciclo.



examples

Deseábamos sacar los números pares entre el numero 500 y el 1000

```
for(int i=500;i<=1000;i+=2)
{//Notemos que escribir i+=2 es similar a escribir i = i + 2
System.out.println(i);
}
```

El código funcional completo de la clase sería el siguiente:

```
public class CicloFor
{
    public static void main(String args[])
    {
        for(int i=500;i<=1000;i+=2)
        {
            System.out.println(i);
        }
    }
}
```

¡Siempre
hacia lo alto!



To practice!

LET'S GO!

¡Siempre
hacia lo alto!



P2Txx: The average height

create a JAVA software that:

Calculates the average height of a group of N children and generated the average

Construir un programa en JAVA donde, calcule la altura promedio de un grupo de N niños y genere el promedio



shutterstock.com • 640298194

¡Siempre
hacia lo alto!



P2Txx: Total saved

create a JAVA software that:

Determines the total saved per person in N months. Yes A person saves X amount of money each month.

The program should calculate the value of total savings and average

Construir un programa en JAVA que, determine el total ahorrado por una persona en N meses. Si Una persona ahorra cada mes una cantidad X de dinero.

El programa deberá calcular el valor del ahorro total y el promedio



P2Txx: Bill of market

create a JAVA software that:

Calculates gross value and total value = gross value + VAT (19%)
for N products

Cree un programa en JAVA que:

calcule el valor_bruto y valor_total = valor_bruto+IVA(19%) para
N productos



P2Txx: salary of N employees

create a JAVA software that:

Calculates the average salary, total salary and highest salary of N employees, where $\text{salary} = (\text{salary} / 30) * \text{working_days}$

crear un programa en JAVA, que:

Calcular el salario promedio, salario total y salario más alto de N empleados, donde $\text{salario} = (\text{sueldo} / 30) * \text{dias_trabajados}$



P2Txx: Tour of a bus

create a JAVA software that:

simulates the route of a bus, at each stop many passengers get on and off, determine how many passengers were transported and how much money was obtained if each ticket is worth \$ 150

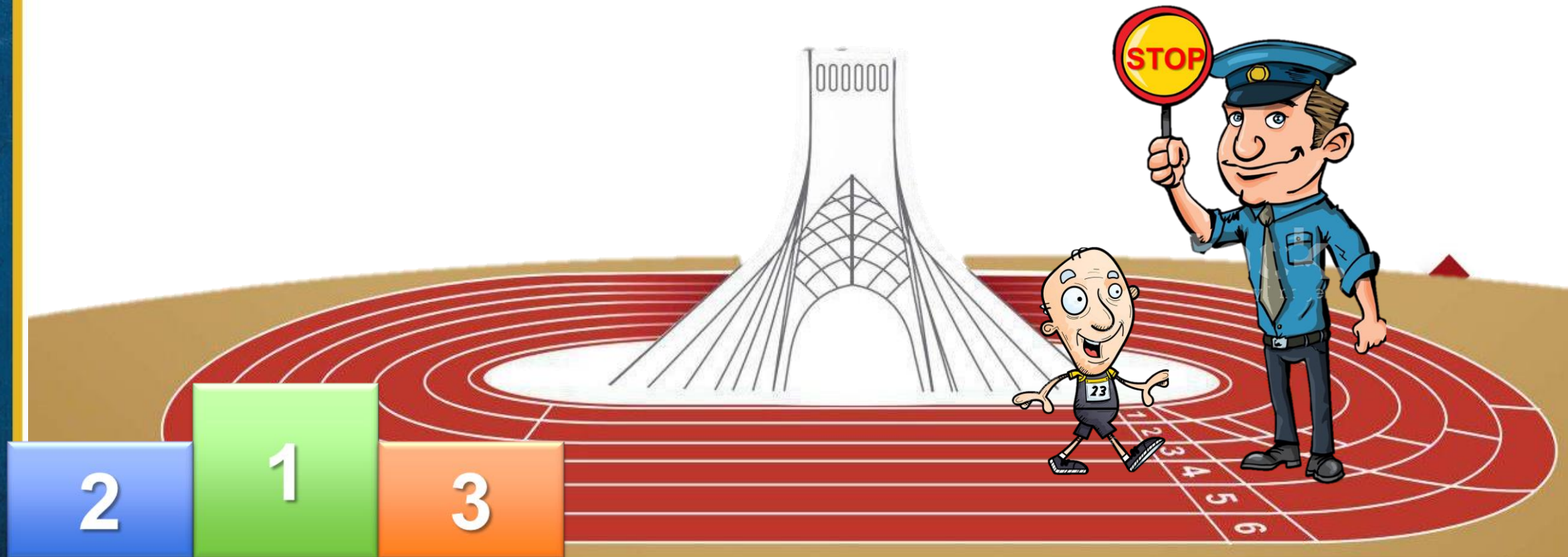
crear un programa en JAVA, que:

simule la ruta de un autobús, en cada parada entran y salen muchos pasajeros, determine cuántos pasajeros fueron transportados y cuánto dinero se obtuvo si cada boleto vale \$ 150



Cycles “WHILE”

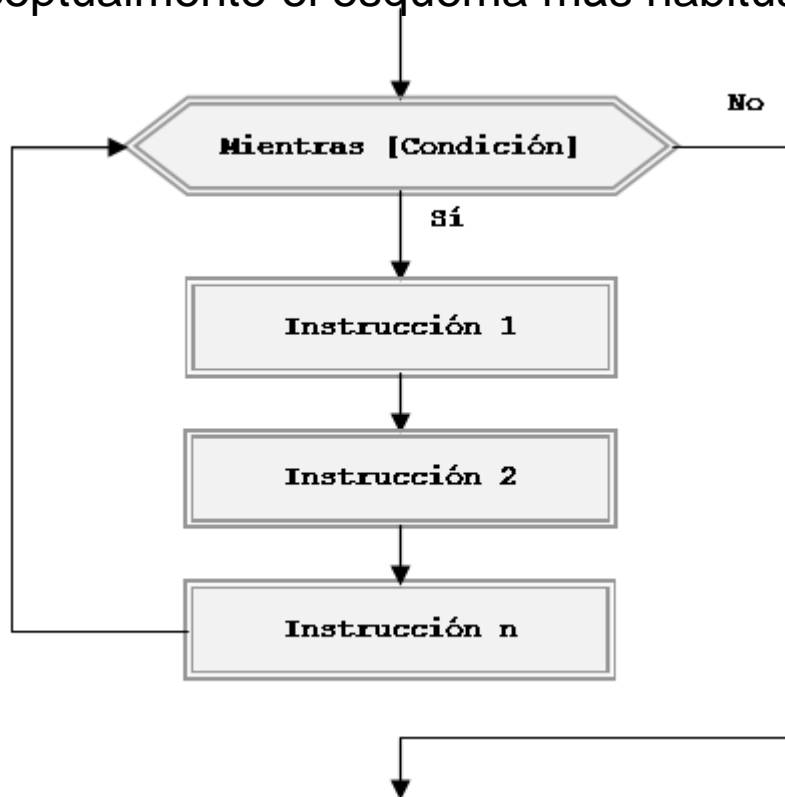
“Hacer lo mismo mientras se pueda”





Syntax

El bucle **WHILE** presenta ciertas similitudes y ciertas diferencias con el bucle **FOR**. La repetición en este caso se produce no un número predeterminado de veces, sino mientras se cumpla una condición. Conceptualmente el esquema más habitual es el siguiente:





Syntax

La sintaxis de un ciclo WHILE es simple en lenguaje JAVA.:

```
1  while (CONDICIÓN)
2  {
3      .....
4      .....
5      Bloque de insrucciones
6      .....
7      .....
8  }
```

- **Línea 1:** definimos la condición con la cual se dará inicio al ciclo y con la que se validara en cada repetición (interacción).
- **Línea 2:** "{" con la cual damos por iniciado el ciclo.
- **Línea 3 a 6:** En estas líneas es donde estarán todo el código fuente que queramos llevar a cabo de manera iterativa durante la ejecución del ciclo
- **Línea 8:** "}" con la cual damos por cerrado el ciclo.

¡Siempre
hacia lo alto!



examples

Creamos un programa que pedirá un número “eternamente” hasta que el número sea diferente de 10.

```
System.out.println("input a number:");
num=keyboard.nextInt();
while (num!=10)
{
    System.out.println("The number is "+num);
    System.out.println("input other number:");
    num=keyboard.nextInt();
}
```

El código funcional completo de la clase sería el siguiente:

```
public class CicloWhile
{
    public static void main(String args[])
    {
        System.out.println("input a number:");
        num=keyboard.nextInt();
        while (num!=10)
        {
            System.out.println("The number is "+num);
            System.out.println("input other number:");
            num=keyboard.nextInt();
        }
    }
}
```

¡Siempre
hacia lo alto!



P2Txx: Different numbers less than 100

create a JAVA software that:

Read from the keyboard a series of numbers until you get one lower than 100, in the end determine the sum and average of the numbers entered

crear un programa en JAVA, que:

Lea desde el teclado una serie de números hasta obtener uno inferior a 100, al final determinar la suma y promedio de los números ingresados.



P2Txx: Different numbers less than 100

create a JAVA software that:

A housewife needs to make the market of the month and only has \$ 500,000 pesos. Perform an algorithm that captures the price of each product and adds it until the account reaches the maximum limit (500,000), in the end determine how many products you buy and when it was the total value you paid

crear un programa en JAVA, que:

Un ama de casa requiere hacer el mercado del mes y solo tiene \$500.000 pesos. Realizar un algoritmo que capture el precio de cada producto y lo sume hasta que la cuenta llegue al tope máximo (500.000), al final determine cuantos productos compro y cuando fue el valor total que pago.



P2Txx: make market with 500 thousand

create a JAVA software that:

A housewife needs to make the market of the month and only has \$ 500,000 pesos. Perform an algorithm that captures the price of each product and adds it until the account reaches the maximum limit (500,000), in the end determine how many products you buy and when it was the total value you paid

crear un programa en JAVA, que:

Un ama de casa requiere hacer el mercado del mes y solo tiene \$500.000 pesos. Realizar un algoritmo que capture el precio de cada producto y lo sume hasta que la cuenta llegue al tope máximo (500.000), al final determine cuantos productos compro y cuando fue el valor total que pago.



P2Txx: sport time

create a JAVA software that:

Records the times of each lap of training and determines the best lap, the average and the number of laps per training (the laps will be until the athlete gets tired / exhausted)

crear un programa en JAVA, que:

Registre los tiempos de cada vuelta de entrenamiento y me determine la mejor vuelta, el promedio y el número de vueltas por entrenamiento (las vueltas serán hasta que se canse/agote el deportista)



P2Txx: password

create a JAVA software that:

Ask for a password (3 attempts maximum) or until the password is 352, 259 or 569.

crear un programa en JAVA, que:

Pedir una contraseña (3 intentos máximo) o hasta que la clave sea 352, 259 ó 569.



¡Siempre
hacia lo alto!



P2Txx: password

create a JAVA software that:

Ask for a password (3 attempts maximum) or until the password is 352, 259 or 569.

crear un programa en JAVA, que:

Pedir una contraseña (3 intentos máximo) o hasta que la clave sea 352, 259 ó 569.



¡Siempre
hacia lo alto!





UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

¡Siempre hacia lo alto!

USTATUNJA.EDU.CO



@santotomastunja