

UNIDAD 3

Programación C#

José Monterroso

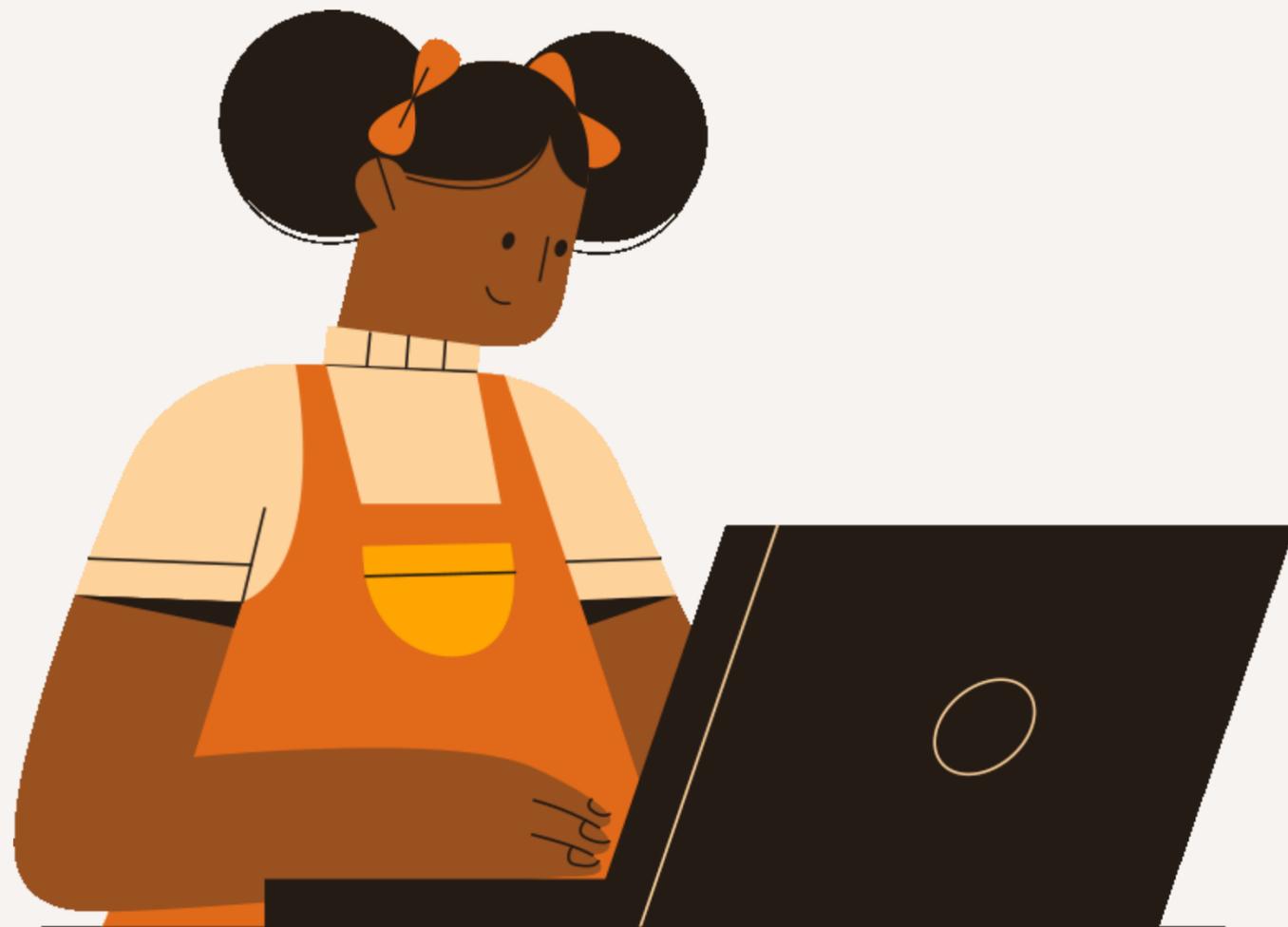


Cadenas de Caracteres

Hemos estado utilizando en todos los problemas planteados desde el principio la definición de una variable de tipo string donde almacenamos cualquier dato que carga el operador por teclado, esto debido a que a clase Console tiene el metodo ReadLine que carga un string.

Mas adelante veremos en profundidad y detenimiento los conceptos del manejo de string, por ahora nos interesa la mecanica para trbajar con cadenas de caracteres.



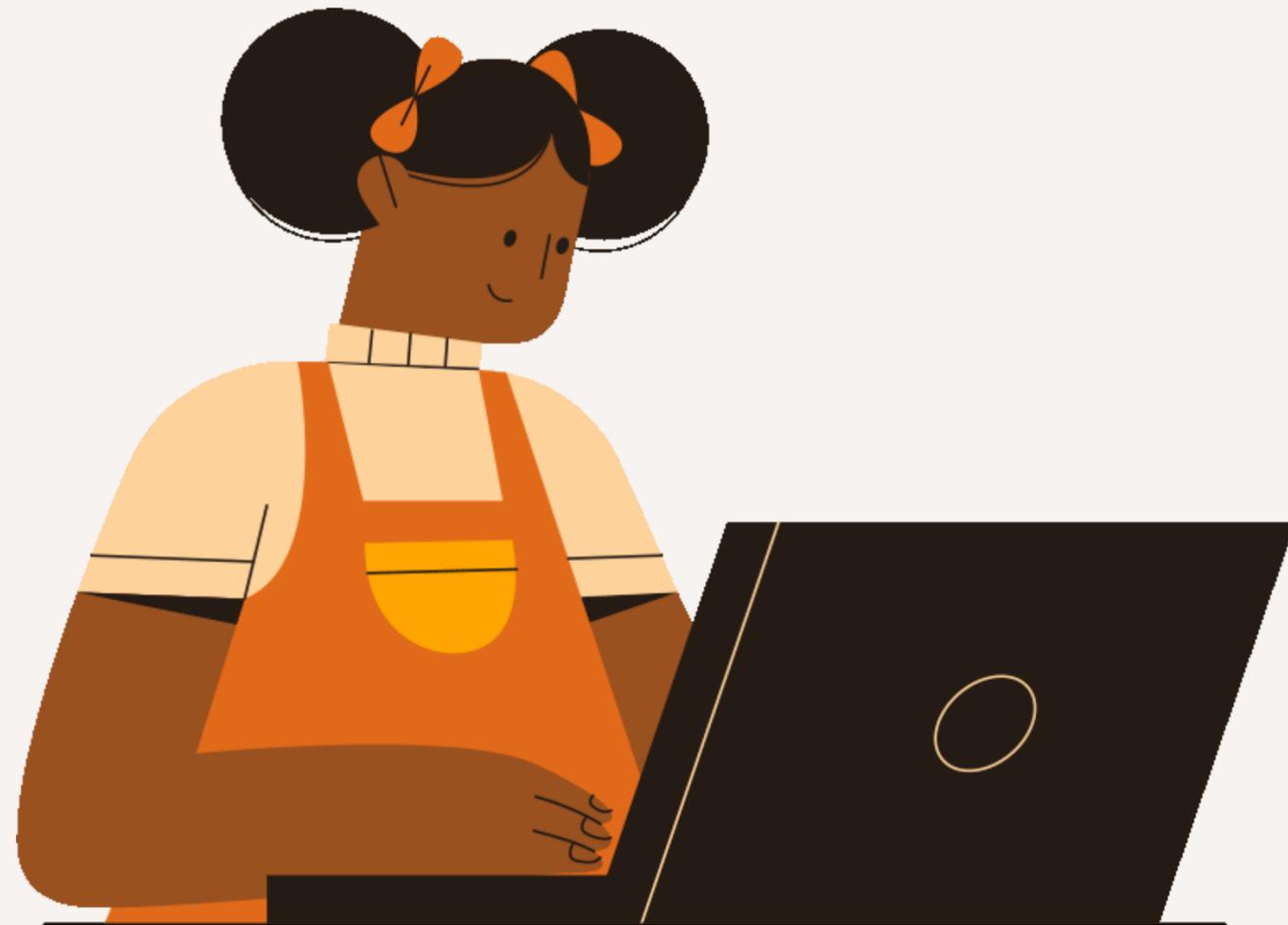


Problema 1

Ejercicio en Clase:

Solicitar el ingreso del nombre y edad de dos personas. Mostrar el nombre de las personas con mayor edad.





Problema 2

Ejercicio en Clase:

Solicitar el ingreso de dos apellidos. Mostrar un mensaje si son iguales o distintos.



Declaración de una clase y definición de objetos.



La programación orientada a objetos se basa en la programación de clases; a diferencia de la programación estructurada, que está centrada en las funciones.

Una clase es un molde del que luego se pueden crear múltiples objetos, con similares características.

Una clase es una plantilla (molde), que define atributos (variables) y métodos (funciones)

La clase define los atributos y métodos comunes a los objetos de ese tipo, pero luego, cada objeto tendrá sus propios valores y compartirán las mismas funciones.

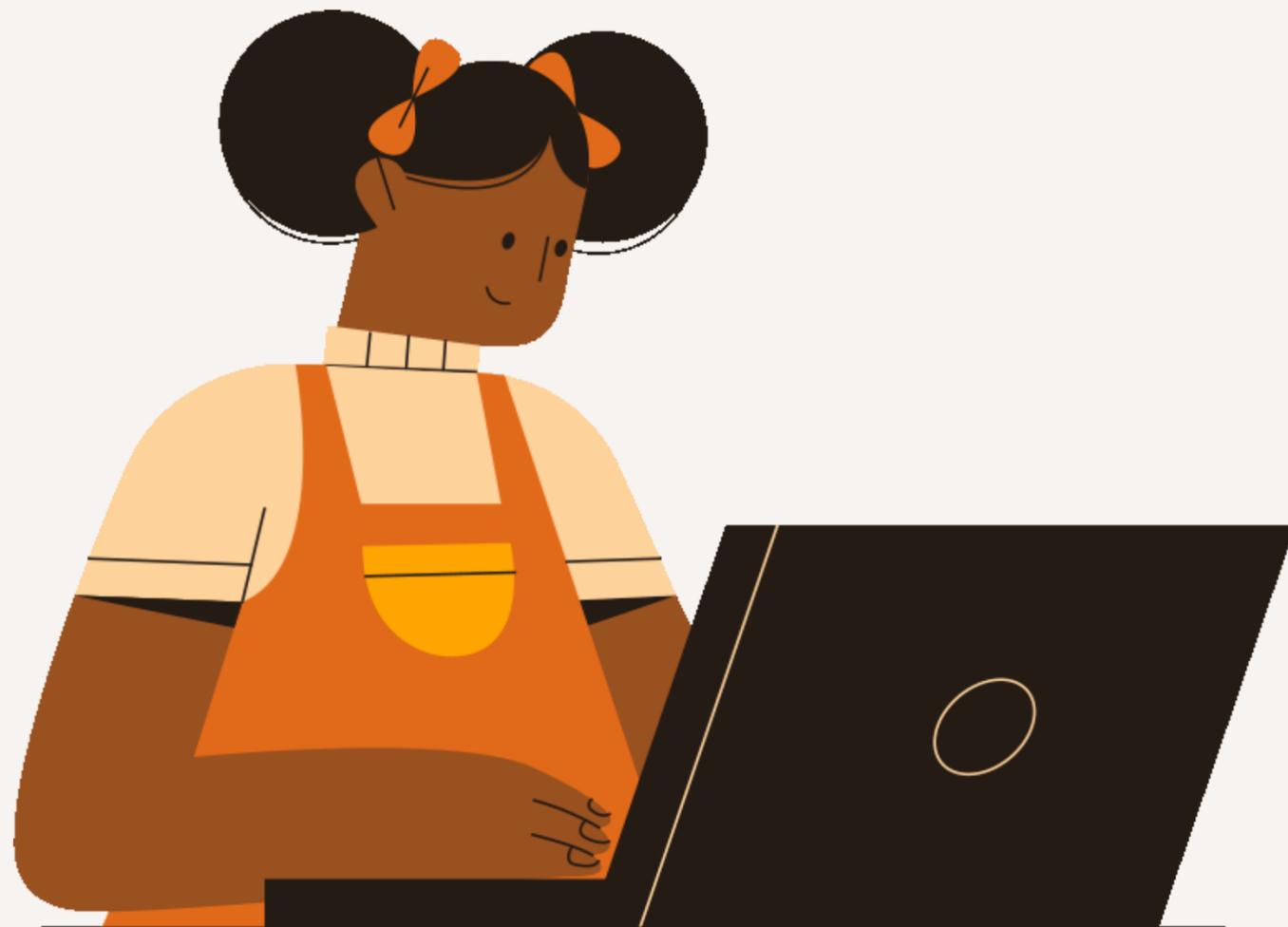


Debemos crear una clase antes de poder crear objetos (instancias) de esa clase. Al crear un objeto de una clase, se dice que se crea una instancia de la clase o un objeto propiamente dicho.

La estructura de una clase es:

```
class [nombre de la clase] {  
  
    [atributos o variables de la clase]  
  
    [métodos o funciones de la clase]  
  
    [main]  
}
```



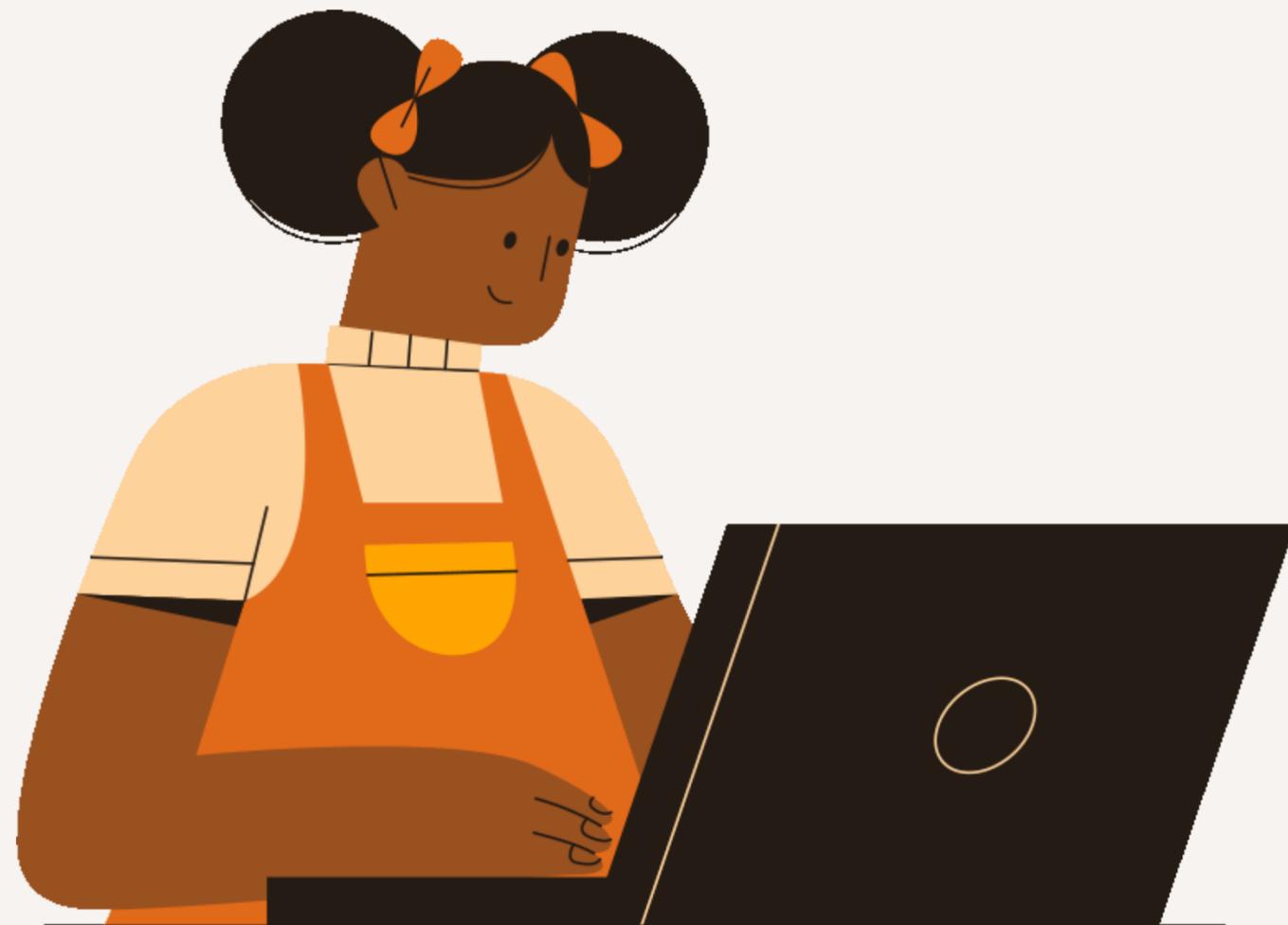


Problema 3

Ejercicio en Clase:

Confeccionar una clase que permita carga el nombre y la edad de una persona. Mostrar los datos cargados. Imprimir si es mayor de edad ($\text{edad} \geq 18$).



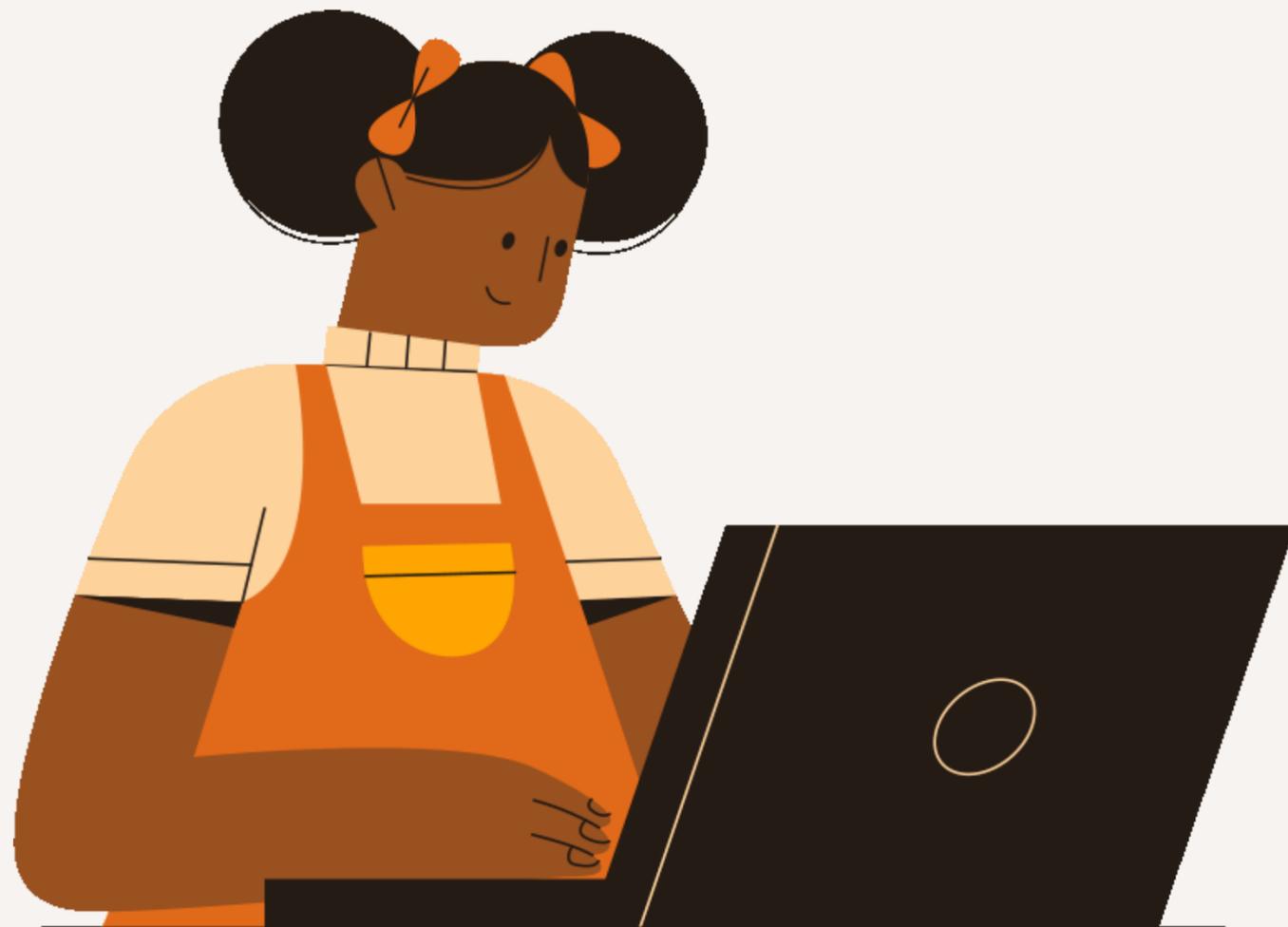


Problema 4

Ejercicio en Clase:

Desarrollar un programa que cargue los lados de un triángulo e implemente los siguientes métodos: solicitar los lados, notificar el valor del lado mayor y otro método que muestre si es equilátero o no.

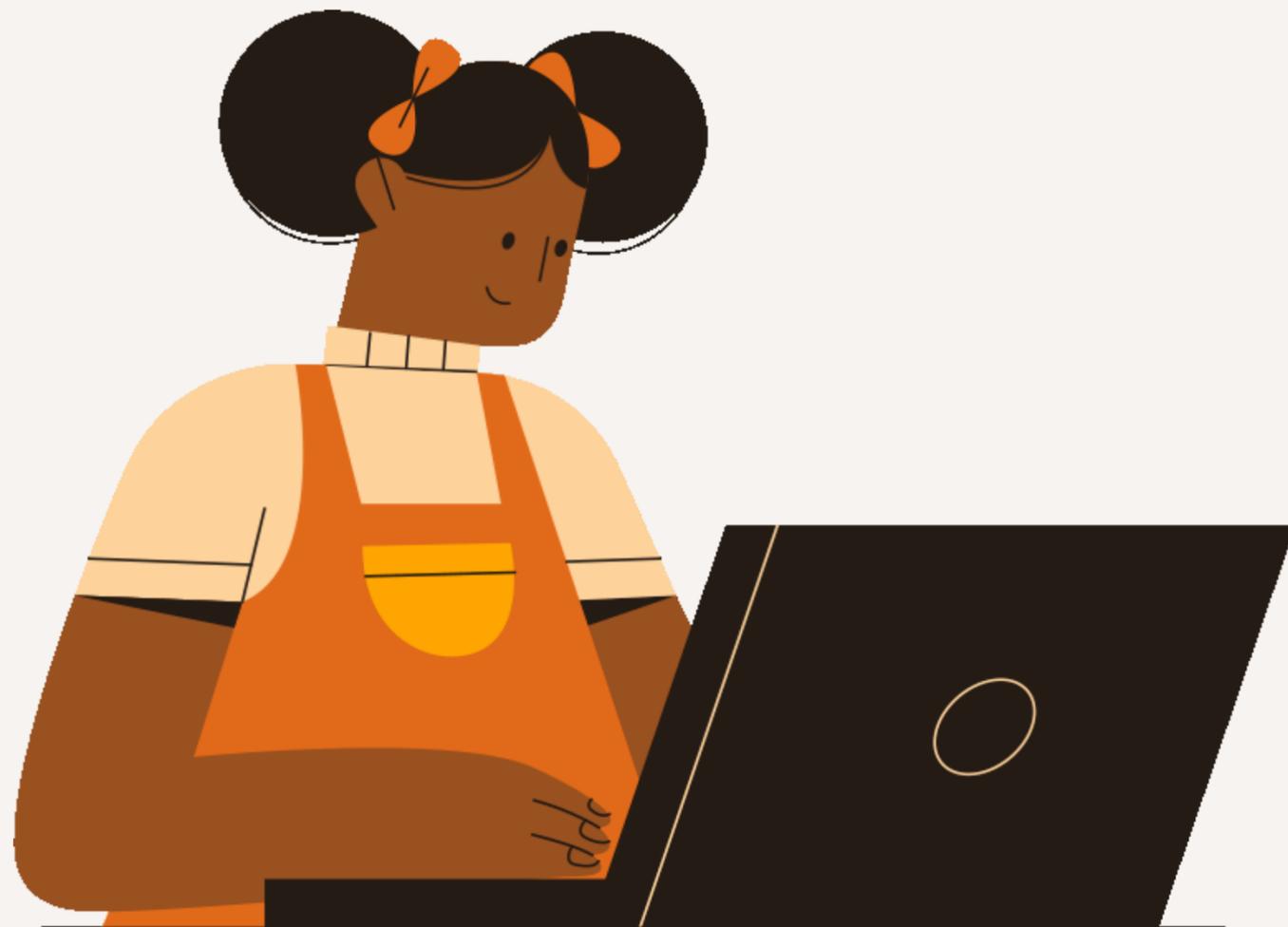




Ejercicio 1

Desarrollar una clase que represente un Cuadrado y tenga los siguientes métodos: cargar el valor de su lado, imprimir su perímetro y su superficie.





Ejercicio 2

Implementar la clase operaciones. Se deben cargar dos valores enteros, calcular su suma, resta, multiplicación y división, cada una en un método, imprimir dichos resultados.



Repetitivo "for"

Cualquier problema que requiera una estructura repetitiva se puede resolver empleando la estructura while. Pero hay otra estructura repetitiva cuyo planteo es más sencillo en ciertas situaciones.

En general, la estructura for se usa en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones. Ejemplo: cargar 10 números, ingresar 5 notas de alumnos, etc. Conocemos de antemano la cantidad de veces que queremos que el bloque se repita.



La variable del for puede tener cualquier nombre. En este ejemplo se la ha definido con el nombre f.

Analicemos el ejemplo:

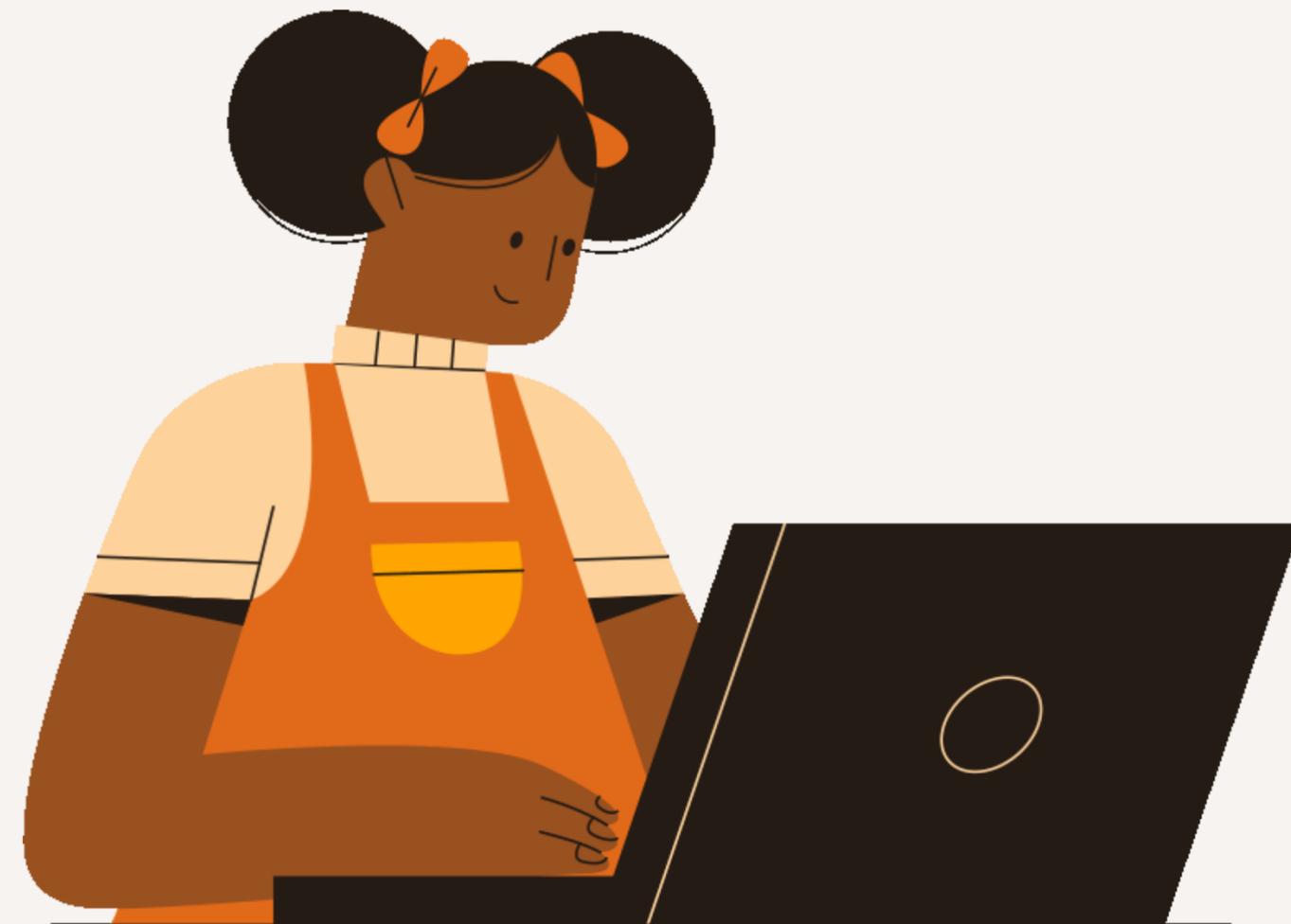
- La variable f toma inicialmente el valor 1.
- Se controla automáticamente el valor de la condición: como f vale 1 y esto es menor que 50, la condición da verdadero.
- Como la condición fue verdadera, se ejecutan la/s operación/es.
- Al finalizar de ejecutarlas, se retorna a la instrucción f++, por lo que la variable f se incrementa en uno.
- Se vuelve a controlar (automáticamente) si f es menor o igual a 50.

Como ahora su valor es 2, se ejecuta nuevamente el bloque de instrucciones e incrementa nuevamente la variable del for al terminar el mismo.

- El proceso se repetirá hasta que la variable f sea incrementada al valor 51.

En este momento la condición será falsa, y el ciclo se detendrá.





Problema 5

Ejercicio en Clase:

Desarrollar un programa que cargue en pantalla los números del 1 al 100.



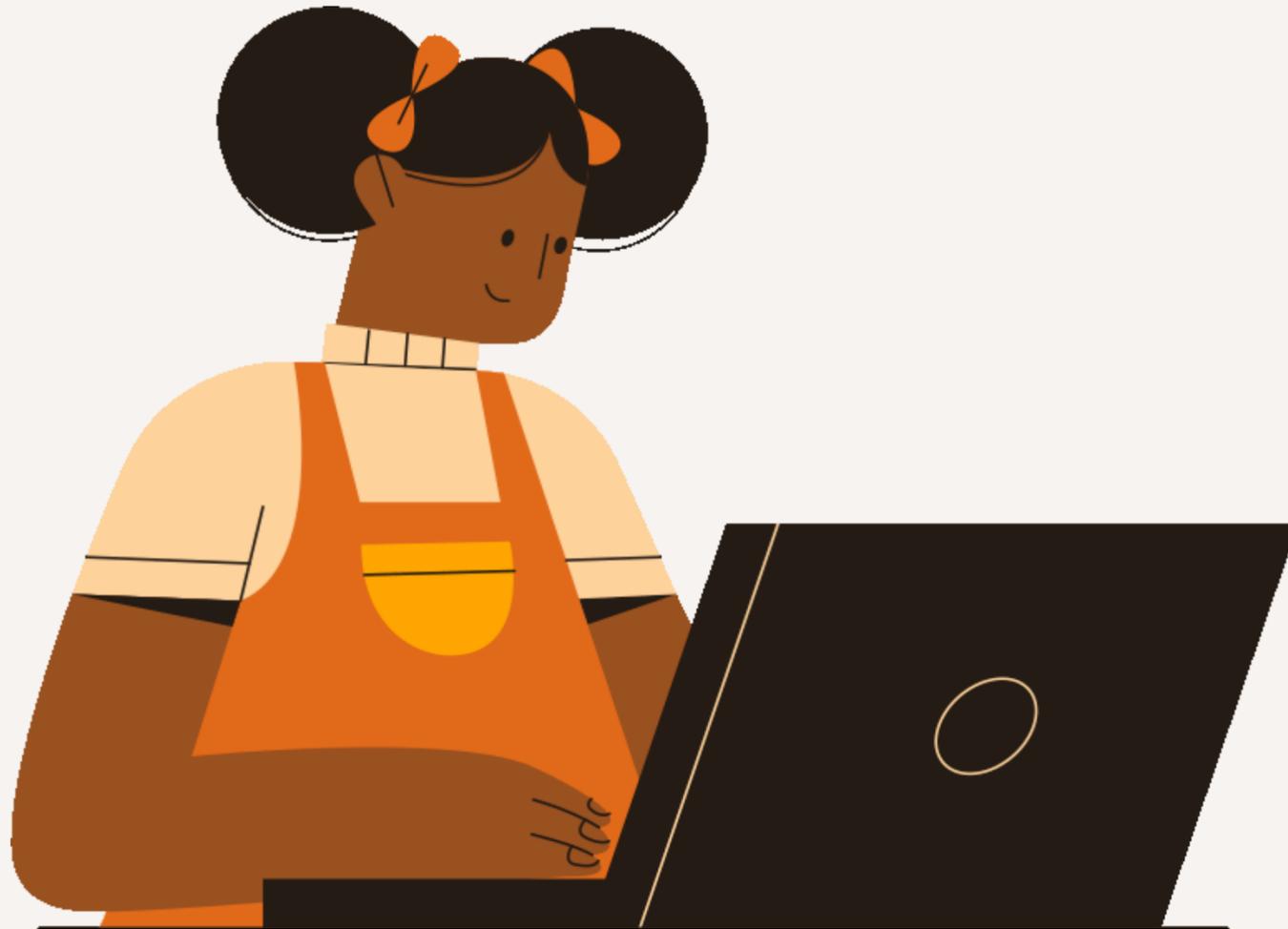
Problema 6

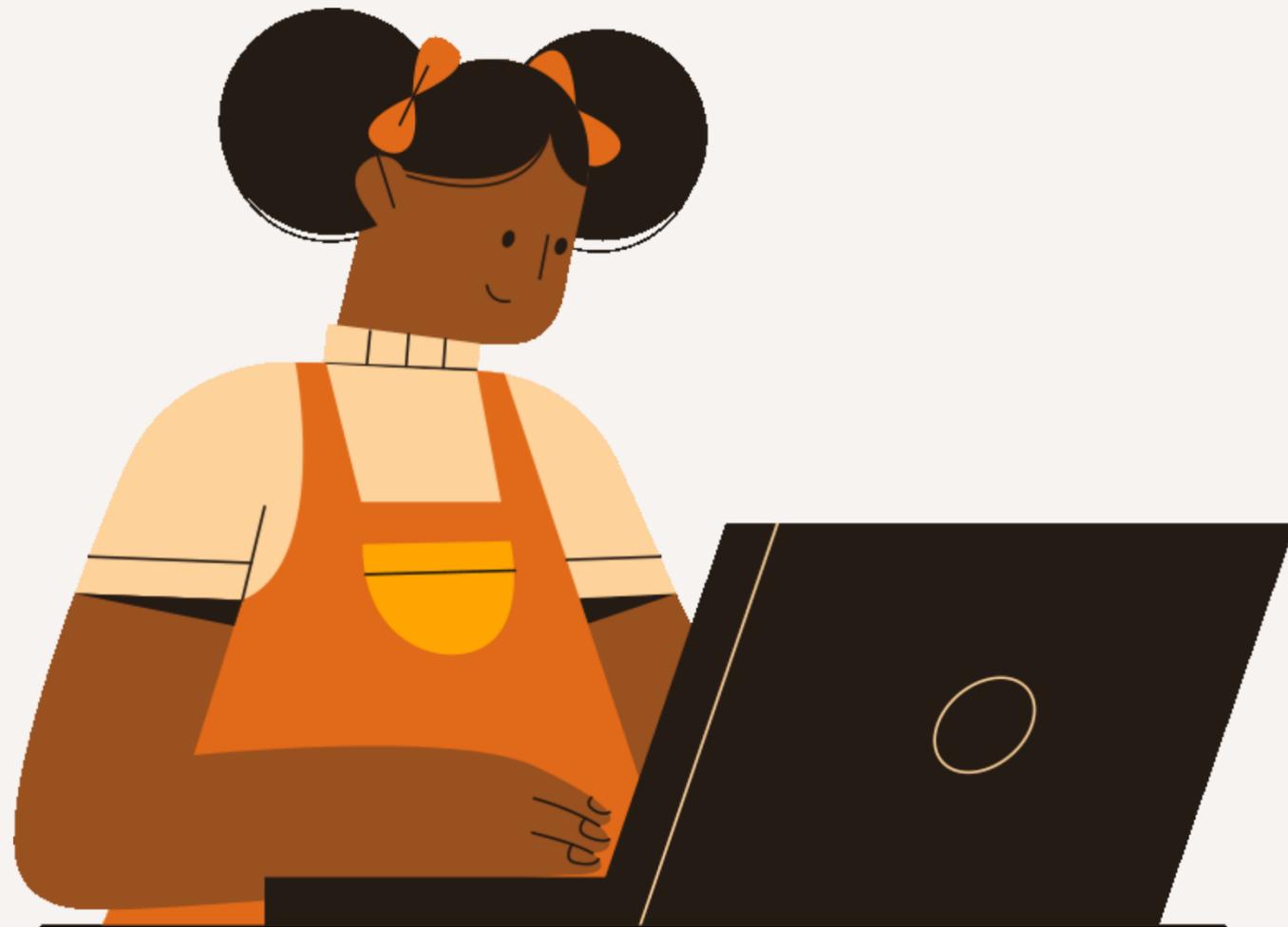
Ejercicio en Clase:

Desarrollar un programa que cargue 10 notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores.

Para resolver este problema se requieren tres contadores:

1. Aprobados (Cuenta la cantidad de alumnos aprobados)
2. Reprobados (Cuenta la cantidad de reprobados)
3. f (es el contador del for)

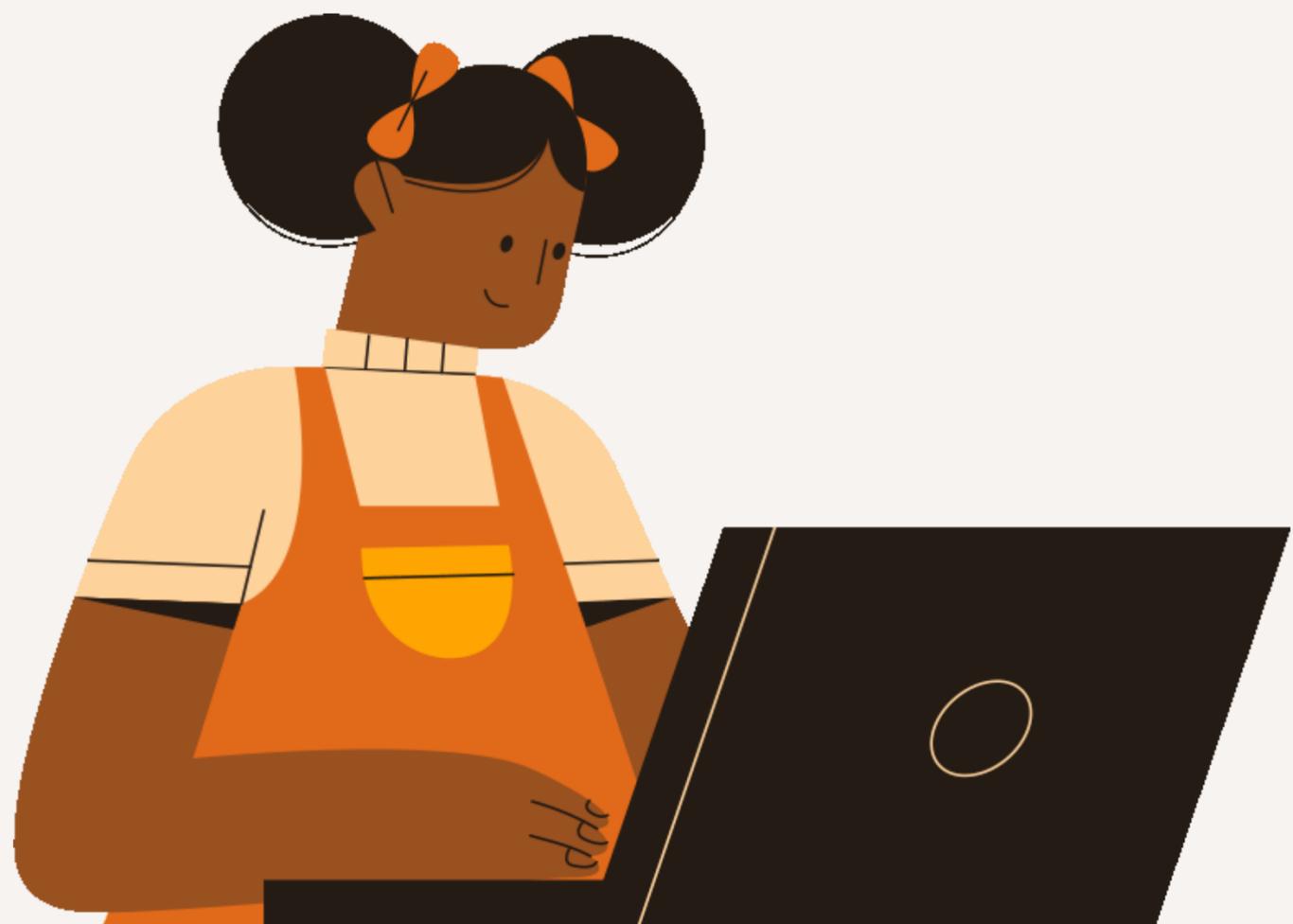




Ejercicio 3

Desarrollar un programa que permita la carga de 10 valores por teclado y nos muestre posteriormente la suma de los valores ingresados y su promedio.

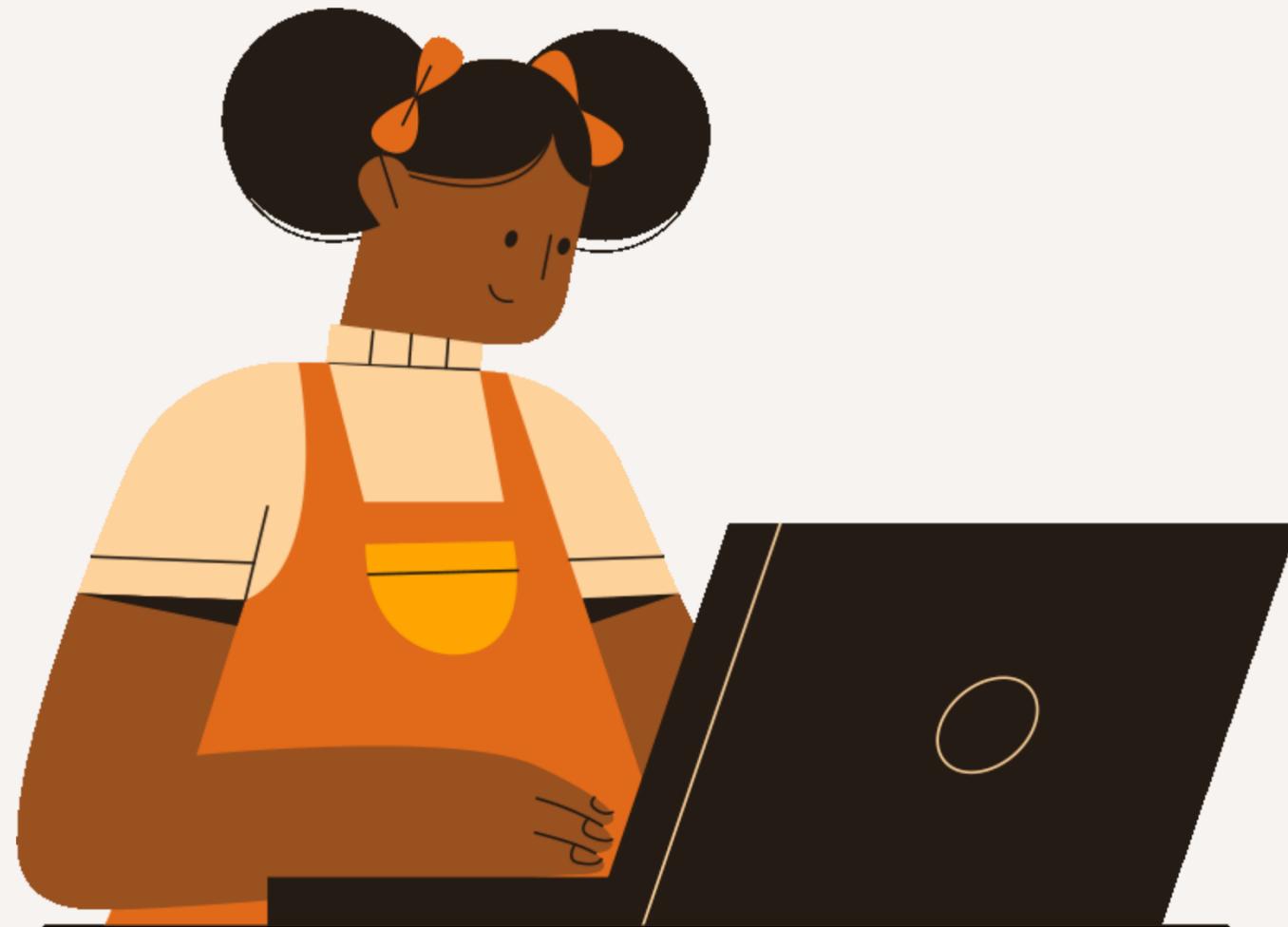




Ejercicio 4

Desarrollar un programa que muestre la tabla de multiplicar del 5 (del 5 al 50).





Ejercicio 5

Desarrollar un programa que solicite la carga de 10 números e imprima la suma de los últimos 5 valores ingresados.



Repetitiva

"do while"

La estructura do while es otra estructura repetitiva, la cual ejecuta al menos una vez su bloque repetitivo, a diferencia del while o del for que podían no ejecutar el bloque.

Esta estructura repetitiva se utiliza cuando conocemos de antemano que por lo menos una vez se ejecutará el bloque repetitivo.

La condición de la estructura está abajo del bloque a repetir, a diferencia del while o del for que está en la parte superior.



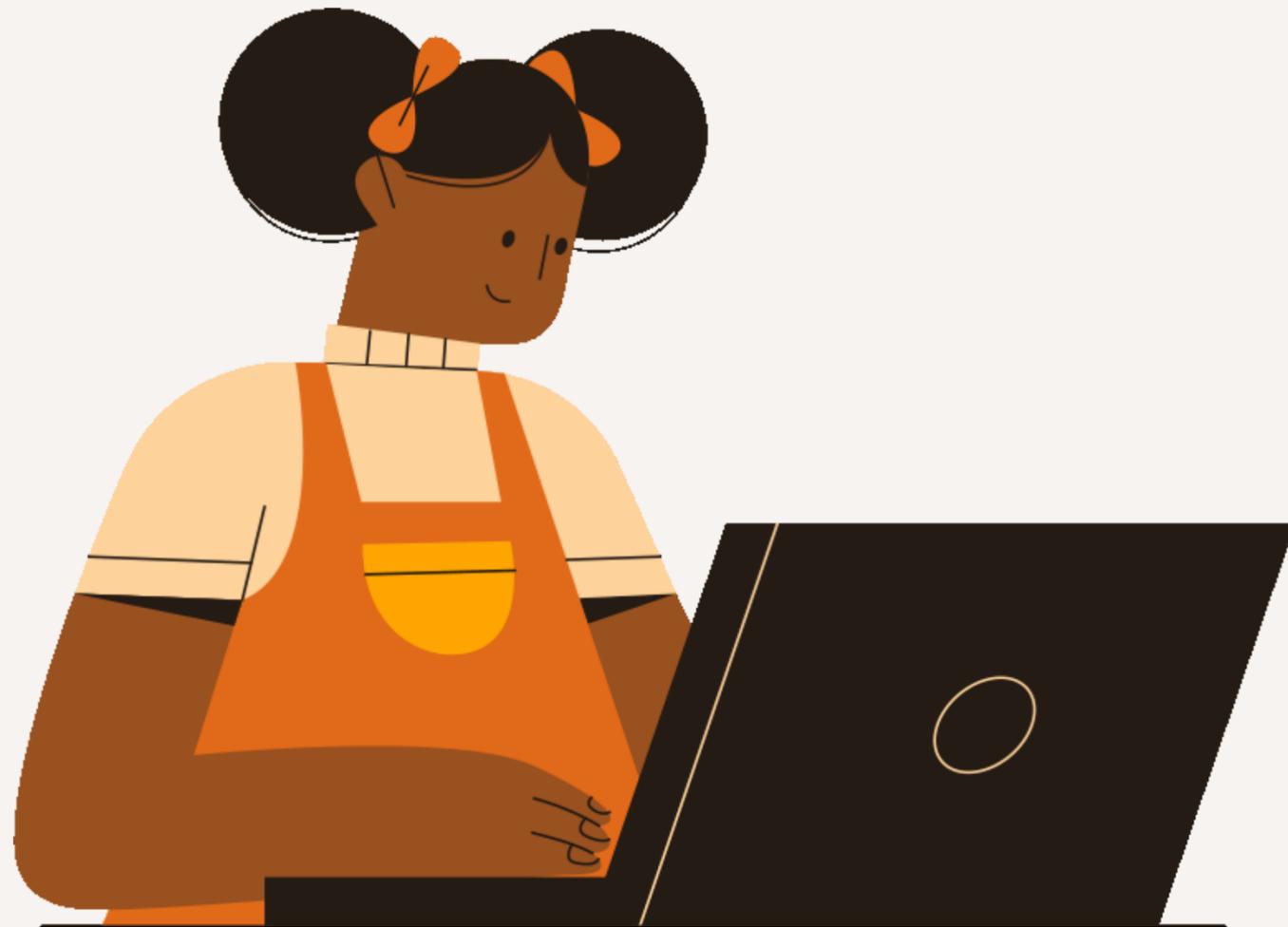


El bloque de operaciones se repite MIENTRAS que la condición sea Verdadera.

Si la condición retorna Falso el ciclo se detiene. En C#, todos los ciclos repiten por verdadero y cortan por falso.

Es importante analizar y ver que las operaciones se ejecutan como mínimo una vez.



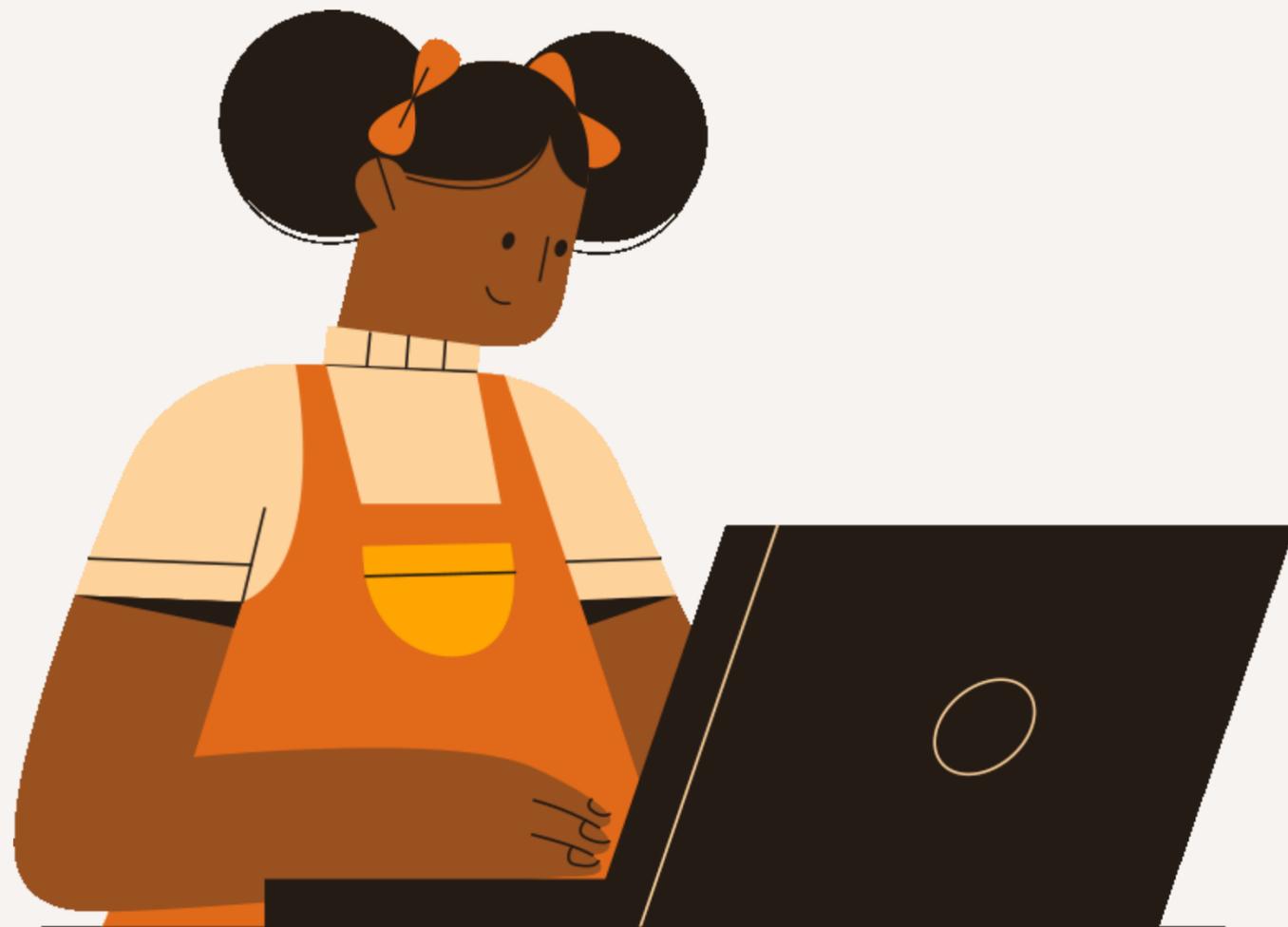


Problema 7

Ejercicio en Clase:

Desarrollar un programa que solicite la carga de un número entre 0 y 999, y nos muestre un mensaje de cuántos dígitos tiene el mismo. Finalizar el programa cuando se cargue el valor 0.





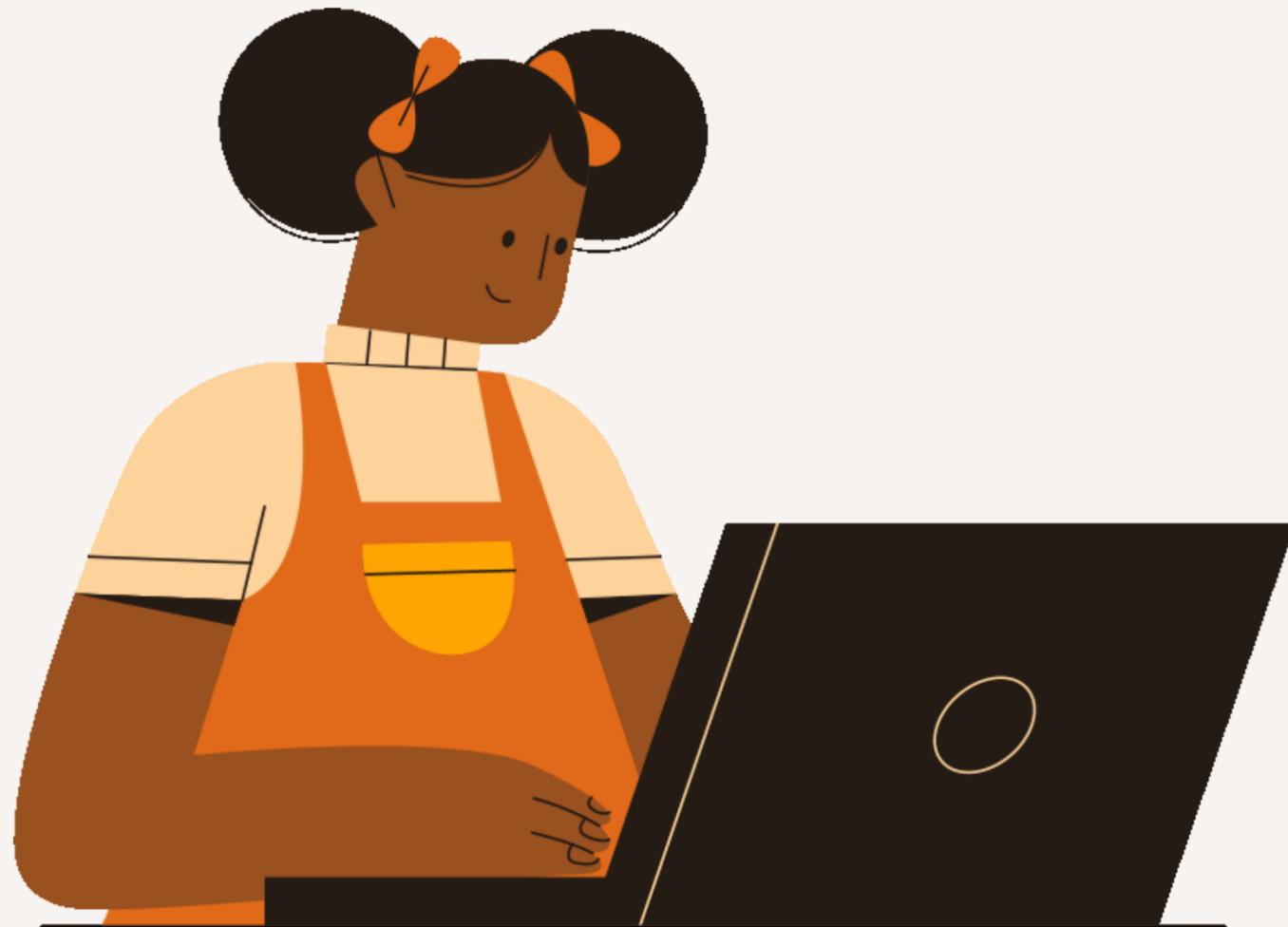
Problema 8

Ejercicio en Clase:

Desarrollar un programa que solicite la carga de números, obtener su promedio. Finalizar la carga de valores cuando se cargue el valor 0.

La finalización depende de algún valor ingresado por el operador empleando de la estructura do while, por lo menos se cargará un valor (en el caso más extremo se carga 0, que indica la finalización de la carga de valores).





Ejercicio 6

Realizar un programa que acumule (sume) valores ingresados por teclado hasta ingresar el 9999 (no sumar dicho valor, indica que ha finalizado la carga). Imprimir el valor acumulado e informar si dicho valor es cero, mayor a cero o menor a cero.



Declaración de métodos

Cuando uno plantea una clase en lugar de especificar todo el algoritmo en un único método (lo que hicimos en los primeros pasos de este tutorial) es dividir todas las responsabilidades de la clase en un conjunto de métodos.

Un método hemos visto que tiene la siguiente sintaxis:

```
public void [nombre del método]()  
{  
    [algoritmo]  
}
```



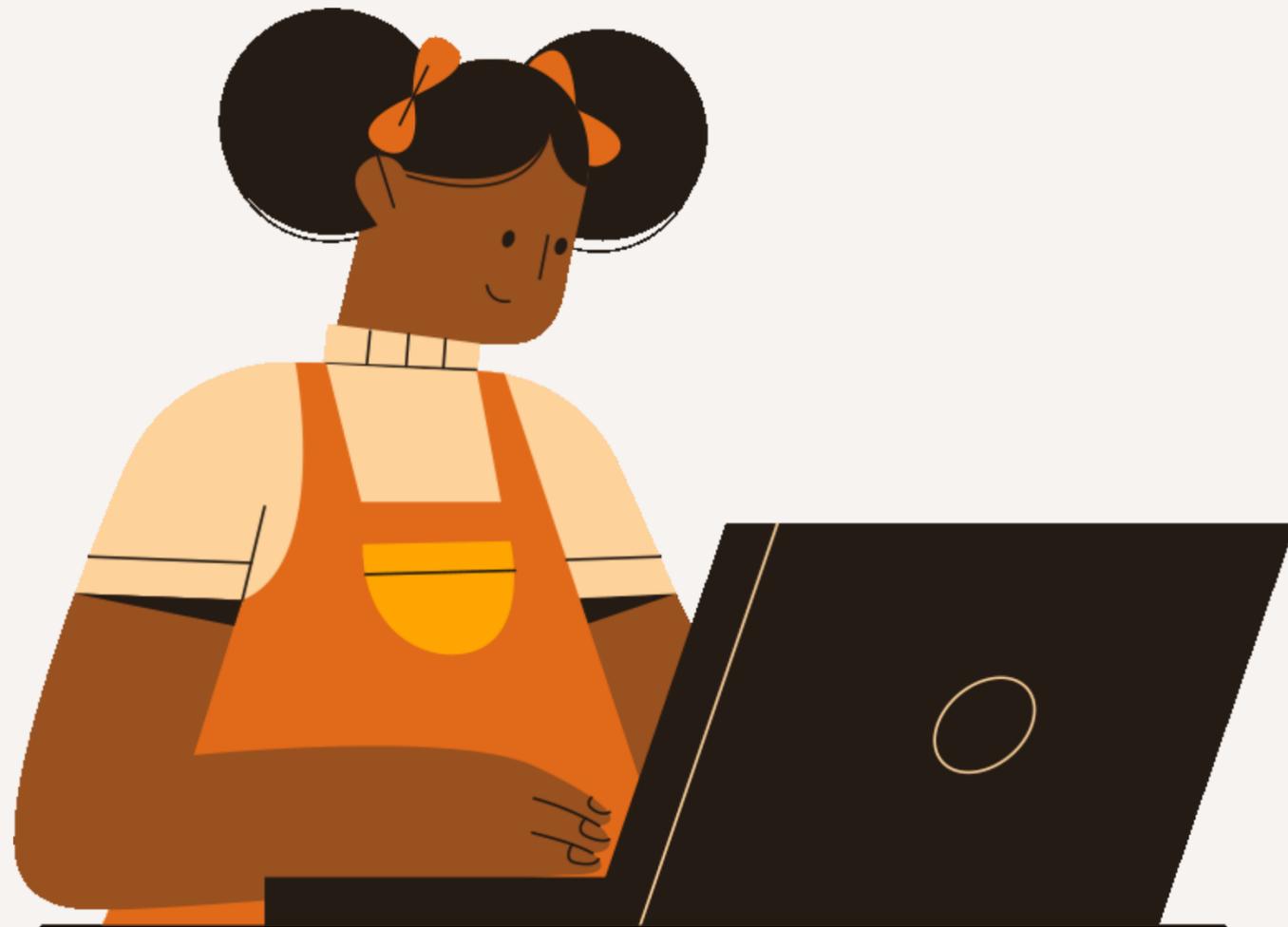
Métodos con parámetros

Un método puede tener parámetros:

```
public void [nombre del método]([parámetros])  
{  
    [algoritmo]  
}
```

Los parámetros los podemos imaginar como variables locales al método, pero su valor se inicializa con datos que llegan cuando lo llamamos.





Problema 9

Ejercicio en Clase:

Desarrollar un programa que confeccione una clase que permita ingresar valores enteros y le muestre al usuario la tabla de multiplicar de dicho valor. Finalizar el programa al ingresar el -1 . (Tabla a multiplicar del 1 al 10)



Métodos que retornan un dato

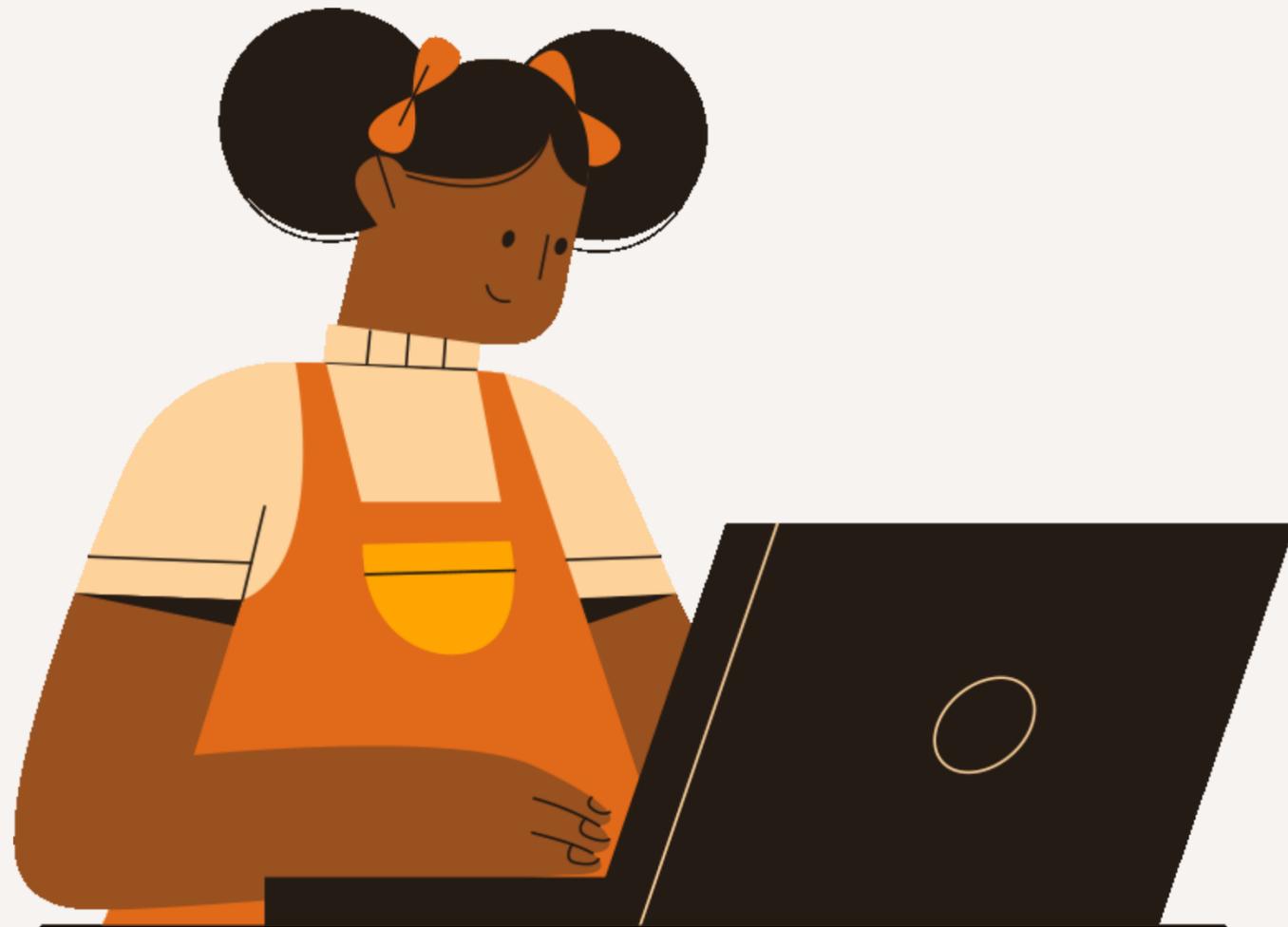


Un método puede retornar un dato:

```
public [tipo de dato] [nombre del método]
([parámetros])
{
    [algoritmo]
    return [tipo de dato]
}
```

Cuando un método retorna un dato en vez de indicar la palabra clave void previo al nombre del método indicamos el tipo de dato que retorna. Luego dentro del algoritmo en el momento que queremos que finalice el mismo y retorne el dato empleamos la palabra clave return con el valor respectivo.

Métodos que retornan un dato

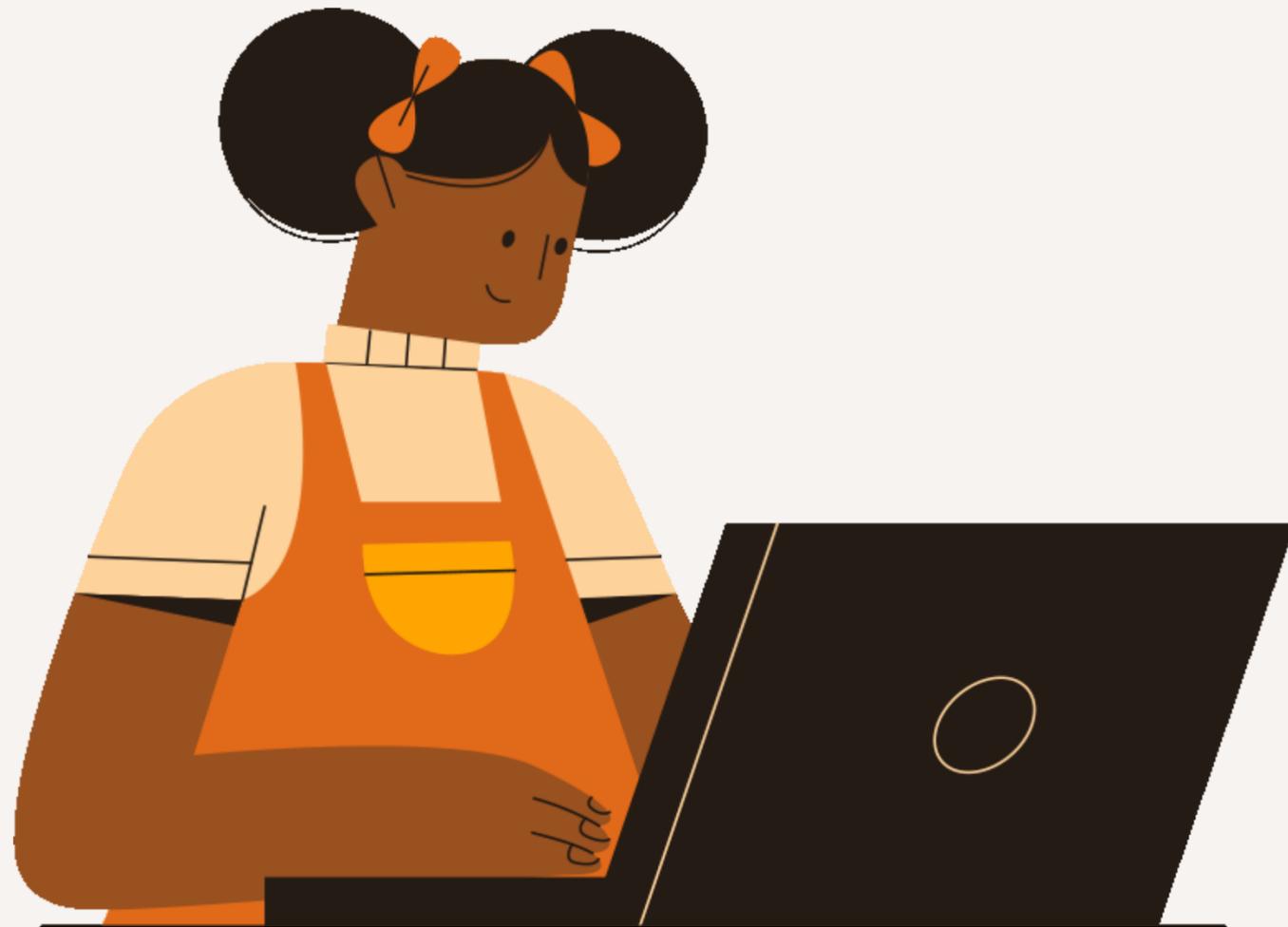


Problema 10

Ejercicio en Clase:

Realizar un programa que indique una clase que permita ingresar tres valores por teclado. Luego mostrar el mayor en un metodo, y luego mostrar el menor en otro metodo y esta información sera enviada al metodo principal.





Problema 11

Ejercicio en Clase:

Realizar un programa que tenga una clase que represente un Cuadrado y tenga los siguientes métodos: ingresar valor a su lado, imprimir su perímetro y su superficie

