C# .Net Programming: A graphical approach
Class 3

Ing. Hazael Fernando Mojica García

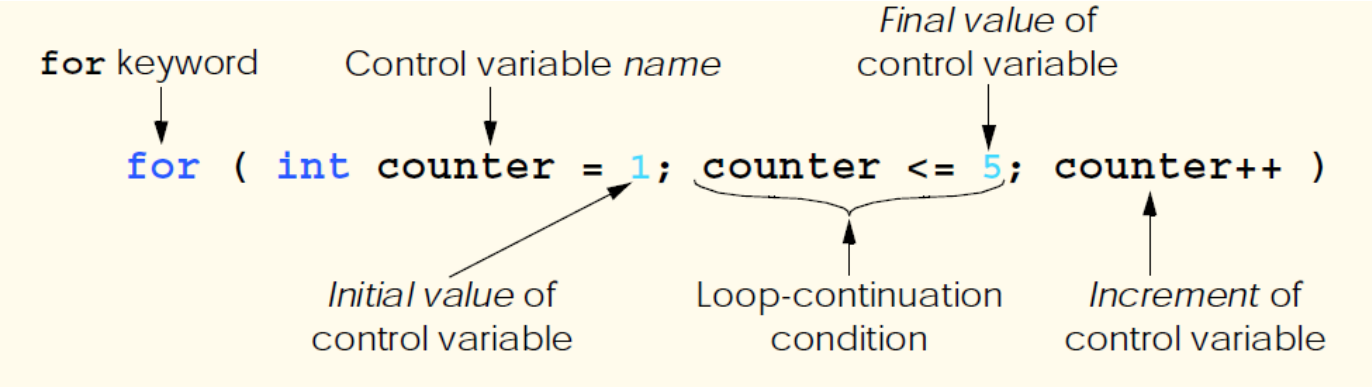
- Flow control structures: *for, switch case*
- *Reserved words: break, continue.*

Flow control sctructure: *for…*

The flow control structure ***for*** allows us to repeat in a definite way a block of code, it's very alike to the while loop but it has the advantage that is easier to read and to find how many times the loop will be repeated.

It is better to use the ***for loop*** when:

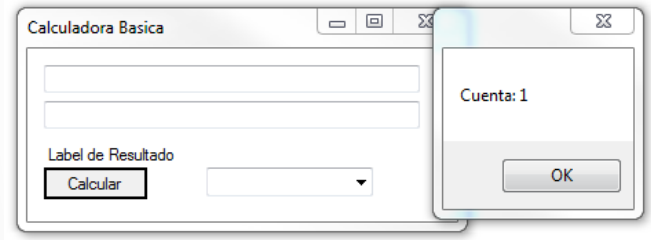
- You now beforehand how many times a loop must be repeated
- You have an array you want to iterate

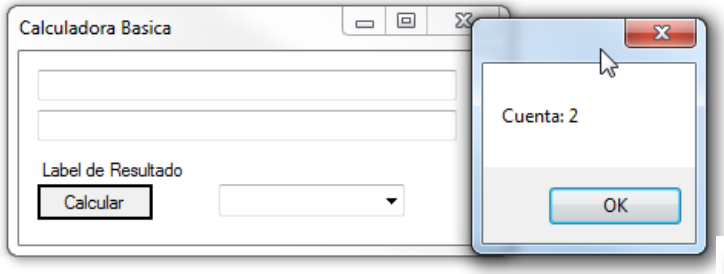


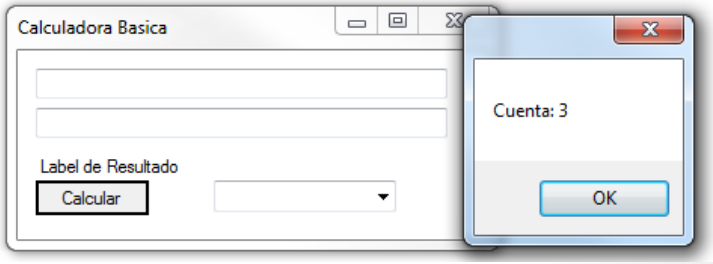

```
int i = 0;
for (i = 1; i <= 10; i++)
{
    //Repeat 10 times
}
```

Using the code we already have for “Calculadora Básica”, add the next block of code to the click event.

```
private void button_Calcular_Click(object sender, EventArgs e)
{
    int i = 0;
    for (i = 1; i <= 3; i++)
    {
        MessageBox.Show(“Count: " + i);
    }
}
```





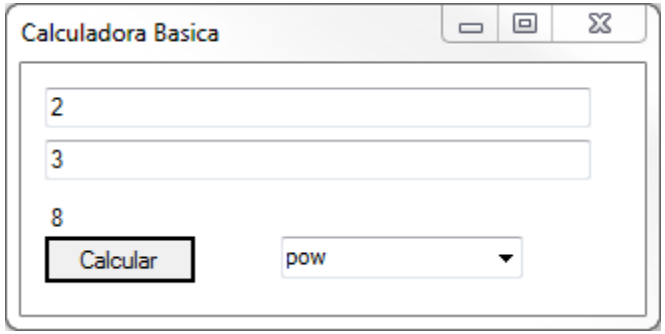


Example 1 *for*

Changing the **pow** function we created, using a ***for loop*** instead of a ***while loop***.

```
else if(operador == "pow")
{//POTENCIA
    int i = 1;
    resultado = valor1;
    while(i < valor2)
    {//Repetir tantas veces como valor2
        resultado = resultado * valor1;
        i++;
    }
}
```

```
else if(operador == "pow")
{//POTENCIA
    int i = 1;
    resultado = valor1;
    for(i = 1; i < valor2; i++)
    {//Repetir tantas veces como valor2
        resultado = resultado * valor1;
    }
}
```



Example 2 *for*
3.1-CalculadoraBasica

Completed code of the method

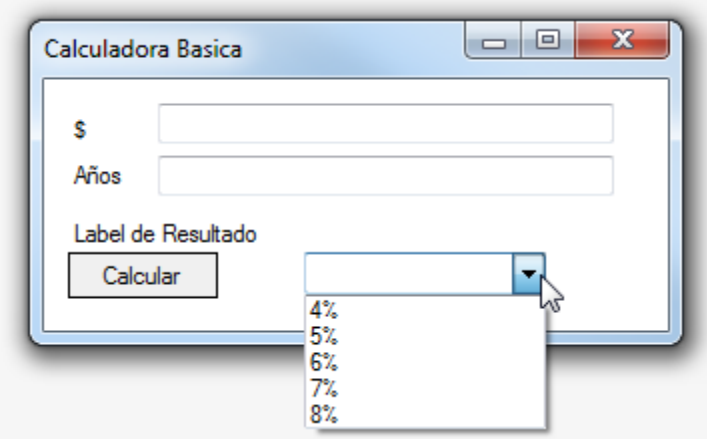
```
    private void button_Calcular_Click(object sender, EventArgs e)
    {
        //Usamos la propiedad Text de la instancia comboBox1 de la clase ComboBox
        //para accesar al valor seleccionado (esto no es robusto pero sirve para nuestro ejemplo)
        String operador = this.comboBox1.Text;
        int resultado = 0;
        int valor1 = 0;
        int valor2 = 0;

        //Obtenemos los valores de los TextBox's
        valor1 = Convert.ToInt32(this.textBox_Val1.Text);
        valor2 = Convert.ToInt32(this.textBox_Val2.Text);

        if (operador == "+")
        {//SUMA
            resultado = valor1 + valor2;
        }
        else if (operador == "-")
        {//RESTA
            resultado = valor1 - valor2;
        }
        else if (operador == "*")
        {//MULTIPLICACION
            resultado = valor1 * valor2;
        }
        else if (operador == "/")
        {//DIVISION
            resultado = valor1 / valor2;
        }
        else if(operador == "pow")
        {//POTENCIA
            int i = 1;
            resultado = valor1;
            for(i = 1; i < valor2; i++)
            {//Repetir tantas veces como valor2
                resultado = resultado * valor1;
            }
        }
        else
        {//SIN OPERADOR
            MessageBox.Show("Selecciona un operador");
        }

        //Colocamos el resultado en el Label
        this.label_Resultado.Text = resultado.ToString();
    }
}
```

Example 3 *for*.

Using the GUI we already have, create a program that calculates the final amount you will get if you put money in a Fixed-term plan for ***n*** years.

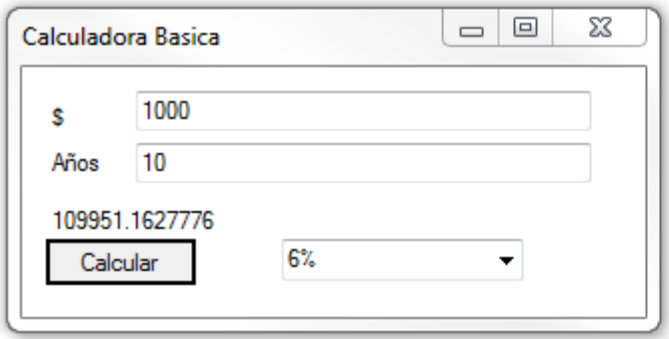
Use the first **TextBox** to type the initial amount of money you want to insert, and the second **TextBox** for the years you want your money in a Fixed-term plan.
Use a **ComboBox** to choose the different earning rates your bank gives you (example 4%, 5%, 6%, 7%, 8%).)

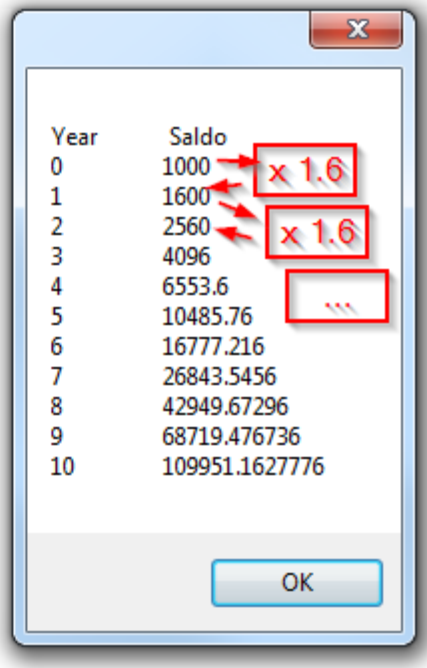
The window would look like:



The code

```
private void button_Calcular_Click(object sender, EventArgs e)
{
    //Obtenemos los datos iniciales
    double net = Convert.ToDouble(this.textBox_Val1.Text);
    double years = Convert.ToDouble(this.textBox_Val2.Text);

    //Seleccionamos el porcentaje de interese
    double p = 0;
    if (this.comboBox1.Text == "4%")
    {
        p = 0.4;
    }
    else if (this.comboBox1.Text == "5%")
    {
        p = 0.5;
    }
    else if (this.comboBox1.Text == "6%")
    {
        p = 0.6;
    }
    else if (this.comboBox1.Text == "7%")
    {
        p = 0.7;
    }
    else if (this.comboBox1.Text == "8%")
    {
        p = 0.8;
    }
    else
    {
        return;//Rompe la ejecucion
    }

    //Creamos una String que contendra el mensaje del desglose
    String text = "Year \t Saldo";
    text += "\n" + 0 + "\t" + net;

    //Realizamos las operaciones.
    int i = 0;
    double ganancia = 0.0;
    for (i = 1; i <= years; i++)
    {//Repetimes el ciclo tantos years se ocupe
        ganancia = net * p;          //Ganancia del year
        net += ganancia;
        text += "\n" + i + "\t" + net;
    }

    //Desplegamos resultados
    this.label_Resultado.Text = net.ToString();
    MessageBox.Show(text);
}
```





The flow control structure: *switch…case*

The flow control structure ***switc…case*** allows us to create a decision tree, you can see this structure as multiple ***if…else***. And if you have many ***if…else*** in your program you can maybe replace them with a ***switc…case*** structure.

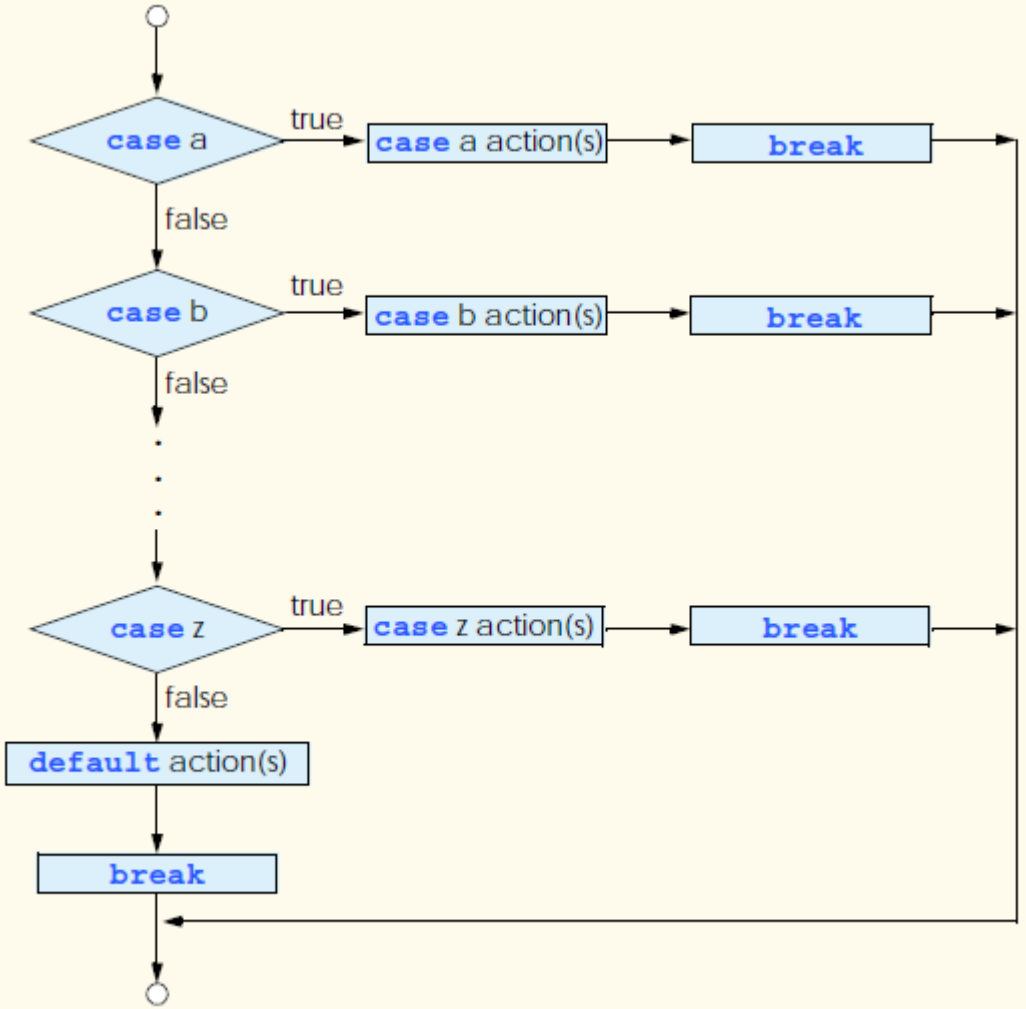
SYNTAX

```
switch (variable_to_compare)
{
    case comparer1:
        //Do something here
        break;

    case comparer2:
        //Do something here
        break;

    case comparer3:
        //Do something here
        break;

    default:
        //Do something here if no condition was true
        break;
}
```

This structure allows us to compare a variable with multiple values, and if the variable match a value (is equal to) then it will execute the block of code after the case sentence until find a **break** keyword.

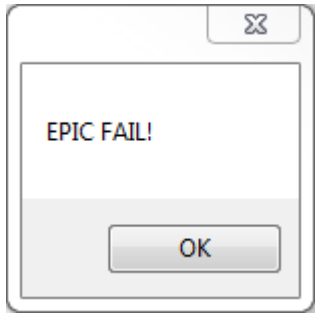
If no condition is evaluated to true then the block of code under the **default:** sentence will be executed.

```
switch (variable)
{
    case "a":
        //Haz algo aqui
        break;

    case “b":
        //Haz algo aqui
        break;

    case “z":
        //Haz algo aqui
        break;

    default:
        //Haz algo aqui
        break;
}
```



Using the existing project or creating a new one copy the next block of code inside a Click event.

```
private void button_Calcular_Click(object sender, EventArgs e)
{
    int numero = 3;

    switch (numero)
    {
        case 6:
            MessageBox.Show("Suerte para la proxima");
            break;

        case 7:
            MessageBox.Show("Ok Pasaste");
            break;

        case 8:
            MessageBox.Show("Not Bad");
            break;

        case 9:
            MessageBox.Show("Good");
            break;

        case 10:
            MessageBox.Show("Nerd");
            break;

        default:
            MessageBox.Show("EPIC FAIL!");
            break;
    }
}
```



Example 1 *switch*

In the code for the project “Calculadora Básica” , replace the ***if…else*** for ***switch.***

```
private void button_Calcular_Click(object sender, EventArgs e)
{
    String operador = this.comboBox1.Text;
    int resultado = 0;
    int valor1 = 0;
    int valor2 = 0;

    //Obtenemos los valores de los TextBox's
    valor1 = Convert.ToInt32(this.textBox_Val1.Text);
    valor2 = Convert.ToInt32(this.textBox_Val2.Text);

    switch (operador)
    {
        case "+"://SUMA
            resultado = valor1 + valor2;
            break;

        case "-"://RESTA
            resultado = valor1 - valor2;
            break;

        case "*"://MULTIPLICACION
            resultado = valor1 * valor2;
            break;

        case "/"://DIVISION
            resultado = valor1 / valor2;
            break;

        case "pow"://POTENCIA
            int i = 1;
            resultado = valor1;
            for (i = 1; i < valor2; i++)
            {//Repetir tantas veces como valor2
                resultado = resultado * valor1;
            }
            break;

        default://SIN OPERADOR
            MessageBox.Show("Selecciona un operador");
            break;
    }

    //Colocamos el resultado en el Label
    this.label_Resultado.Text = resultado.ToString();
}
```

Ejemplo 2 *switch*

Keywords ***break*** and ***continue***

The reserved word ***break*** was used previously inside the ***switch*** structure to delimit the blocks of code.
But it can also be used inside a ***for*** or ***while*** loop to break the iterations and continue with the flow of the program.

The reserved word ***continue*** is only used inside the code block of a ***for*** or ***while*** loop, it tells the program to jump to next iteration.

Syntax for the *break* keyword

```
for (int i = 0; i < 10; i++)
{
    //This code will be executed correctly

    break;

    //This code is unreachable

    i++;
}
```

Is there anything wrong here?

```
int i = 0;
while (i < 10)
{

//This code will be executed correctly

    break;

 //This code is unreachable

    i++;
}
```

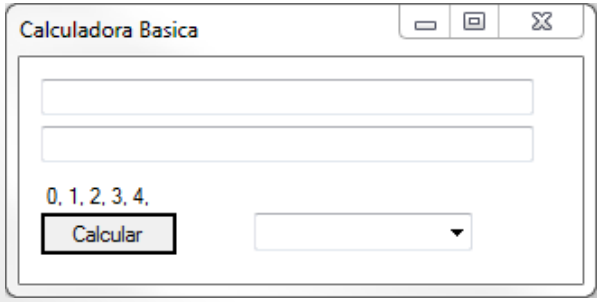
Syntax of *continue*

```
for (int i = 0; i < 10; i++)
{
    //Este codigo es ejecutado

    continue;

    //Este codigo ya no se ejecutara

    i++;
}
```

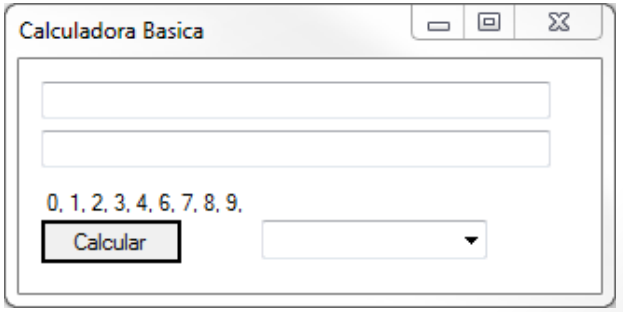
```
int i = 0;
while (i < 10)
{
    //Este codigo es ejecutado

    continue;

    //Este codigo ya no se ejecutara

    i++;
}
```

Examples

```
private void button_Calcular_Click(object sender,
EventArgs e)
{
    String text = "";
    for (int i = 0; i < 10; i++)
    {
        if (i == 5)
            break;
        text += i + ", ";
    }
    this.label_Resultado.Text = text;
}
```




```
private void button_Calcular_Click(object sender,
EventArgs e)
{
    String text = "";
    for (int i = 0; i < 10; i++)
    {
        if (i == 5)
            continue;
        text += i + ", ";
    }
    this.label_Resultado.Text = text;
}
}
```



Logic and conditional **operators**

The **AND - &&** operator

expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

The full sentences is true only of all the conditions are true.

Syntax

```
if (condition1 && condition2 && condition3)
{
    //Execute this block of code only if
    //all the conditions are true
}
```

Example

```
int age = 40;
String gender = "M";
if (age > 25 && gender == "M")
{
    //This code will be executed
}
```

The **OR -** **||** operator

expression1	expression2	expression1 || expression2
false	false	false
false	true	true
true	false	true
true	true	true

The full sentence is true if at least one of the conditions is true

Syntax

```
if (condition1 || condition2 || condition3)
{
    //Execute this block of code if at
    //least one of the conditions is true
}
```

Example

```
int age = 40;
String gender = "M";
if (age > 25 || gender == "F")
{
    //This code won't be executed
}
```

Is there anything wrong here?

The **NOT - !** operator

expression	!expression
false	true
true	false

It will reverse the logical value of the sentence.
If it was true now it will be false, and if it was false now it will be true.

SINTAXIS

```
if ( !expression )
{
    //Este código será ejecutado si
   //expresion es false
}
```

EJEMPLO

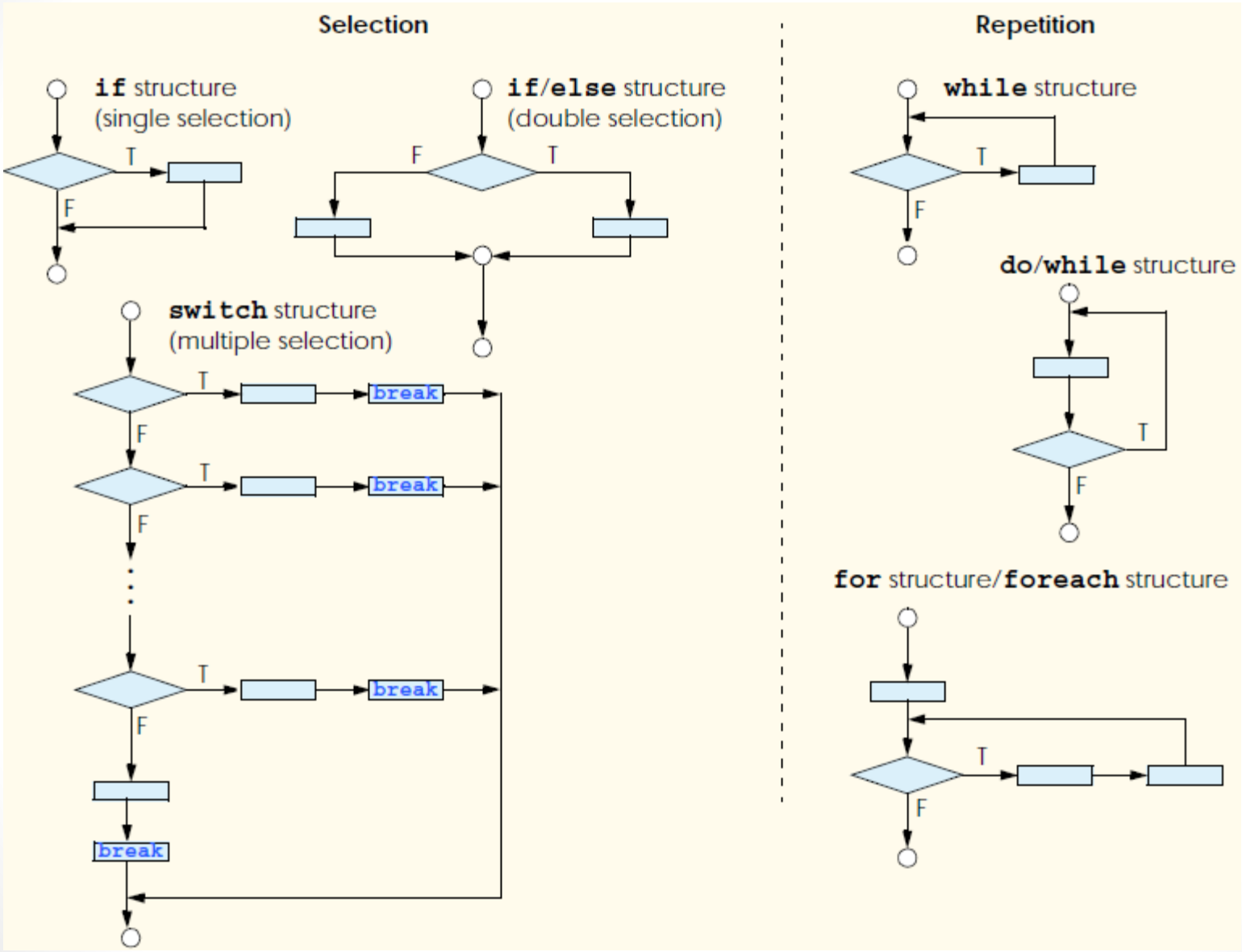
```
String gender = "M";
if ( !(gender == "M") )
{
    //Este código NO será Ejecutado
}
```

```
String gender = "M";
if (gender != "M" )
{
    //Este código NO será Ejecutado
}
```

SUMMARY - Operators

Operators	Associativity	Type
`()`	left to right	parentheses
`++ --`	right to left	unary postfix
`++ -- + - ! (`*type*`)`	right to left	unary prefix
`* / %`	left to right	multiplicative
`+ -`	left to right	additive
`< <= > >=`	left to right	relational
`== !=`	left to right	equality
`&`	left to right	logical AND
`^`	left to right	logical exclusive OR
`|`	left to right	logical inclusive OR
`&&`	left to right	conditional AND
`||`	left to right	conditional OR
`?:`	right to left	conditional
`= += -= *= /= %=`	right to left	assignment

SUMMARY – Flow control structures



Homework

Create a program (you can use the project we already have for “calculadora básica”) that creates a pretty and cute draw of a Christmas tree.
The Christmas tree will be defined by 2 variables: ***height*** and ***figure***.

The user will be able to enter a numeric value for the height using a TextBox (or another similar control) and to choose a figure using a ComboBox, the figures will be at least: “*”, “-”, “%”, “&”.

The Christmas tree will be “painted” in a Message Box when the user click a button.

You should end up with something like: