

Examen final.



Diplomado de Java - Módulo I

Departamento de Diplomados y Extensión Profesional

Centro de Investigación en Computación

Instituto Politécnico Nacional

Modulo I – Examen Final

Profesor:

Alan Badillo Salas

Integrantes:

Erick Alvarez Barcena

becker7000@outlook.es

Fecha de entrega:

30 de abril de 2022

--

--

--

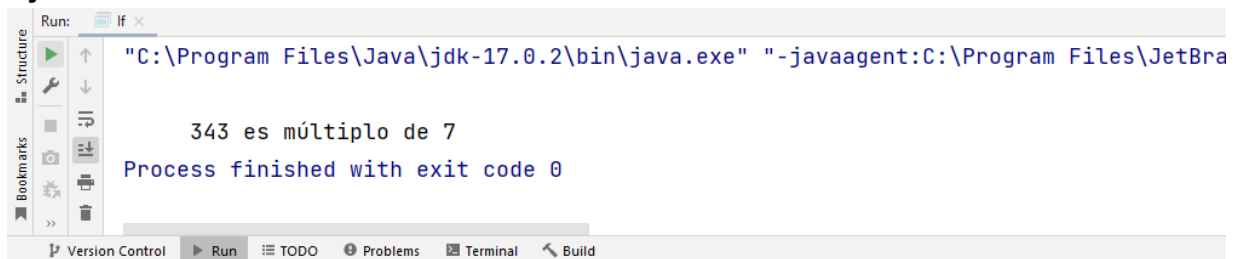
1. Escribe un ejemplo de uso de la estructura condicional if.

En éste ejemplo se declara e inicializa en 343 una variable (**num**) de tipo entero, la cual va a calcularse su módulo 7 (**num%7**), en caso de que el resultado de la operación sea exactamente **0** la comparación entre la expresión a la izquierda del operador relacional igual que (**==**) y la expresión su derecha devuelve un valor booleano verdadero (**true**) éste valor activa el código que está dentro de la estructura **if**, en caso contrario se emite un valor booleano falso (**false**) y el código dentro de la estructura **if** se omite.

Código java.

```
1 public class If {
2     public static void main(String[] args) {
3         //Declaración e inicialización de una variable.
4         int num = 343; //Variable de tipo entero.
5         if(num%7==0) //Comparación de dos expresiones.
6             System.out.printf("\n\t %d es múltiplo de 7",num);
7         //En la línea 9 se muestra en pantalla un mensaje en caso de
8         //que la expresión dentro del parentesis sea verdadera.
9     }
10 }
11
```

Ejecución en IntelliJ IDEA.



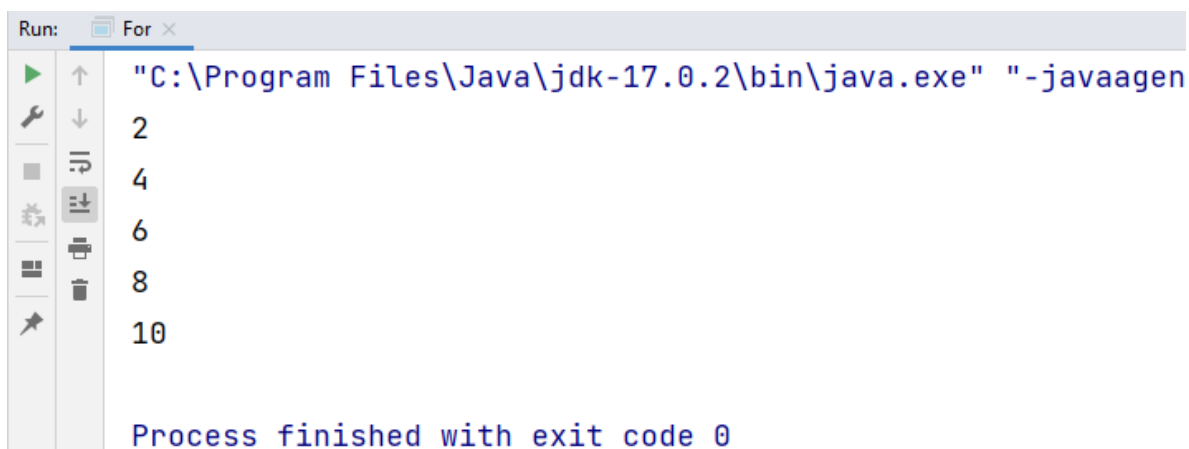
2. Escribe un ejemplo de uso de la estructura del ciclo iterador for.

En el siguiente ejemplo se crea directamente una estructura de control de repetición, éste nos permite de forma compacta declarar su propia variable de control dentro de sus parámetros, la cual inicia en 1, cada parámetro debe ir separado por el signo “;”, después en el parámetro de en medio hay una expresión relacional que en caso de ser verdadera se itera el ciclo, es decir, se ejecuta el código de la línea 8, una vez que se ejecuta el código se incrementa el valor de *i* que se vuelve a evaluar en la expresión relacional.

Código java.

```
1 public class For {
2     public static void main(String[] args) {
3         //Se crea la estructura for
4         //El primer parametro es una variable de control, inicia en 1.
5         //El segundo parametro es una expresión relacional que
6         //para cada vez que se cumpla se ejecutará el código dentro del ciclo.
7         //El tercer parametro es un incremento.
8         for (int i = 1; i <= 5; i++) {
9             System.out.println(i*2);
10        }
11    }
12 }
```

Ejecución en IntelliJ IDEA.



```
Run: For x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagen
2
4
6
8
10

Process finished with exit code 0
```

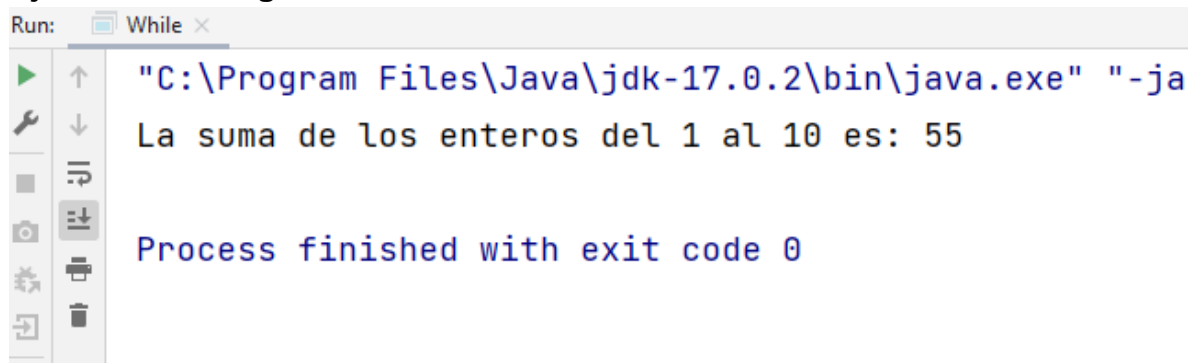
3. Escribe un ejemplo de uso de la estructura ciclo condicional while.

En el siguiente ejemplo se ilustra la suma de Gauss usando la estructura de control **while** de forma simple, primero declaramos e inicializamos en 0 una variable **suma** de tipo entero, esta variable habrá de almacenar las sumas parciales, 1, 1+2, 1+2+3,... luego declaramos otra variable de tipo entera llamada **contador** y se inicia en 1, esta variable va tomando los valores enteros 1,2,3,4,... al empezar el ciclo **while** se evalúa la expresión relacional **contador <=10** la cual devuelve o un valor verdadero o un valor falso, en caso de devolver un valor verdadero entra en la ejecución el bloque de código que contiene dos instrucciones, la primera acumula la suma de los valores que va tomando la variable **contador** y la segunda incrementa en 1 el valor de **contador** para así volver a evaluar la expresión relacional en la siguiente iteración, al cumplirse el ciclo será porque la expresión relacional toma un valor falso y entonces se ejecuta la siguiente instrucción que muestra un mensaje en la consola, éste mensaje dice que resultado final de la suma: **55**.

Código java.

```
1 public class While {
2     public static void main(String[] args) {
3         //Variable que acumula la suma
4         int suma=0;
5         //Variable que toma valores enteros
6         int contador=1;
7         //Mientras el contador sea menor o igual a 100
8         while(contador<=10){ //Inicia el ciclo while
9             suma+=contador; //Instrucción 1
10            contador++; //Instrucción 2
11        }
12        //Se muestra en consola un mensaje.
13        System.out.println("La suma de los enteros del 1 al 10 es: "+suma);
14    }
15 }
```

Ejecutando código en IntelliJ IDEA.



```
Run: While x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-ja
La suma de los enteros del 1 al 10 es: 55
Process finished with exit code 0
```

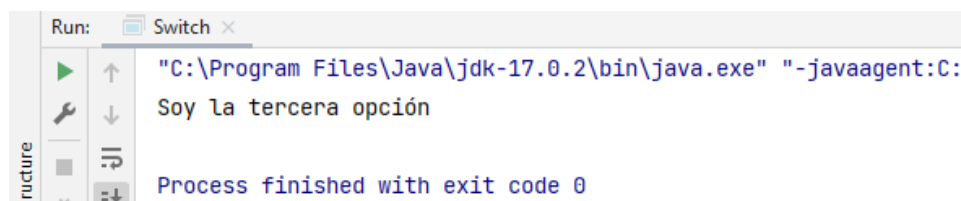
4. Escribe un ejemplo de uso de la estructura selector condicional switch.

En el siguiente ejemplos se emplea una variable de control llamada **opcion** de tipo entera que inicia en **3**, esta variable es pasada como parámetro de la estructura **switch** la cual a partir del valor de opción ejecuta uno de los **case** contemplados, si **opcion** vale **1** ejecuta **case 1**, si opción vale **2** ejecuta **case 2** y así sucesivamente... al finalizar el código dentro del **case** ejecutado se escapa de la estructura general switch con la sentencia **break**, también hemos contemplado un comodín en caso de que la variable **opcion** no contenga el valor de ninguno de los **case** contemplados ese comodín es **default** que al ejecutarse su código se muestra un mensaje en consola, este último no requiere **break** dado que lo que sigue de él es inmediatamente el final de la estructura **switch**.

Código java.

```
1 public class Switch {
2     public static void main(String[] args) {
3
4         int opcion=3; //Variable de control
5
6         switch (3){ //Se recibe la variable de control
7             case 1: //Cada case tiene un único valor constante entero
8                 //Código a ejecutar
9                 System.out.println("Soy la primera opción");
10                break; //Una vez ejecutado el código se escapa de
11                //la estructura switch con la sentencia break;
12            case 2:
13                System.out.println("Soy la segunda opción");
14                break;
15            case 3:
16                System.out.println("Soy la tercera opción");
17                break;
18            default: //default ejecuta un código de comodín
19                // en caso de que el valor de la opcion no entre
20                // en alguno de los caso contemplados.
21                System.out.println("Soy una opción no contemplada");
22        }
23    }
24 }
```

Ejecutando código un IntelliJ IDEA.



Run: Switch x

"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-17.0.2\bin\javaagent.jar" -classpath C:\Users\user\IdeaProjects\Switch\out\production\Switch Switch

Soy la tercera opción

Process finished with exit code 0

5. Generar una clase contador aplicando las siguientes definiciones:

1. Debe poseer un atributo privado de tipo entero llamado `conteo`
2. Debe poseer un constructor público que reciba el valor inicial del atributo `conteo`
3. Debes inicializar el atributo `conteo` al valor inicial dado en el constructor
4. Debes definir un método público que no devuelve nada y no recibe parámetros llamado `incrementar`
5. En el método `incrementar` debes incrementar el atributo `conteo` en `1`
6. Debes definir un método público que no devuelve nada y no recibe parámetros llamado `decrementar`
7. En el método `decrementar` debes decrementar el atributo `conteo` en `1`
8. Debes definir un método público que devuelve un entero y no recibe parámetros llamado `getConteo`
9. En el método `getConteo` debes devolver el valor del atributo `conteo`
10. Debes generar un punto de ejecución (un método estático main)
11. Crea dos contadores y has diversos incrementos y decrementos, imprimiendo el valor devuelto por `getConteo`

* Especifica en la impresión el contador que estás imprimiendo. Ejemplo: `C1: 15` o `C2: -7`

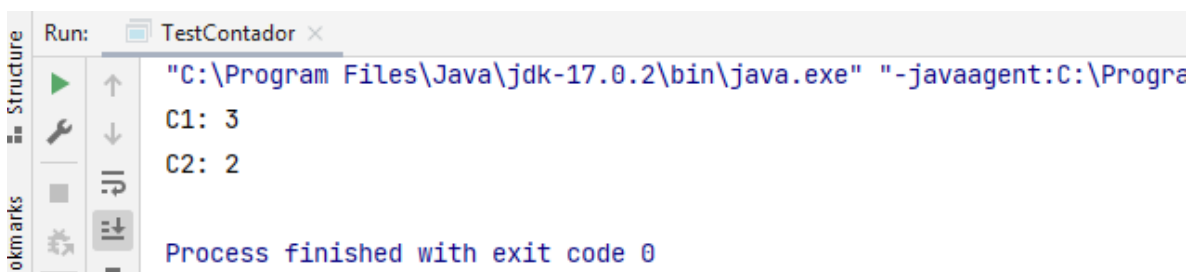
Código java de la clase Contador.

```
1 public class Contador {
2
3     //Variable conteo de tipo entero
4     private int conteo; //Atributo privado.
5
6     //Método constructor que inicializa el valor de la variable conteo.
7     public Contador(int conteo) {
8         this.conteo = conteo; //Se asigna el valor de entrada conteo
9     } // a la variable interna this.conteo
10
11     public void incrementar(){ //Método que no devuelve ningun valor.
12         conteo++; //pero modifica la variable conteo sumandole 1.
13     }
14
15     public void decrementar(){ ///Método que no devuelve ningun valor.
16         conteo--; //pero modifica la variable conteo restandole 1
17     }
18
19     public int getConteo() { //Método de tipo getter
20         return conteo; // devuelve el valor de conteo. (Sólo lectura)
21     }
22
23 }
```

Código java de la clase TestContador (probando la clase Contador).

```
1 public class TestContador {
2     public static void main(String[] args) {
3
4         //Se crean dos objetos (C1,C2) de la clase Contador
5         Contador C1 = new Contador( conteo: 0);
6         Contador C2 = new Contador( conteo: 0);
7
8         //Aplicamos el método incrementar() y decrementar()
9         //algunas veces al objeto C1 y C2:
10
11         C1.incrementar(); C1.incrementar();
12         C1.decrementar(); C1.incrementar();
13         C1.incrementar();
14
15         C2.incrementar(); C2.decrementar();
16         C2.incrementar(); C2.incrementar();
17
18         //Imprimimos el valor final de la variable conteo de cada objeto
19         //Esto lo logramos gracias al método getConteo()
20         System.out.println("C1: "+C1.getConteo());
21         System.out.println("C2: "+C2.getConteo());
22
23     }
24 }
```

Ejecutando la clase con punto de ejecución TestContador.



Run: TestContador x

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Progra
```

C1: 3
C2: 2

Process finished with exit code 0