

Stacks & Queues y Matrices

Heriberto Espino Montelongo

Universidad de las Américas Puebla

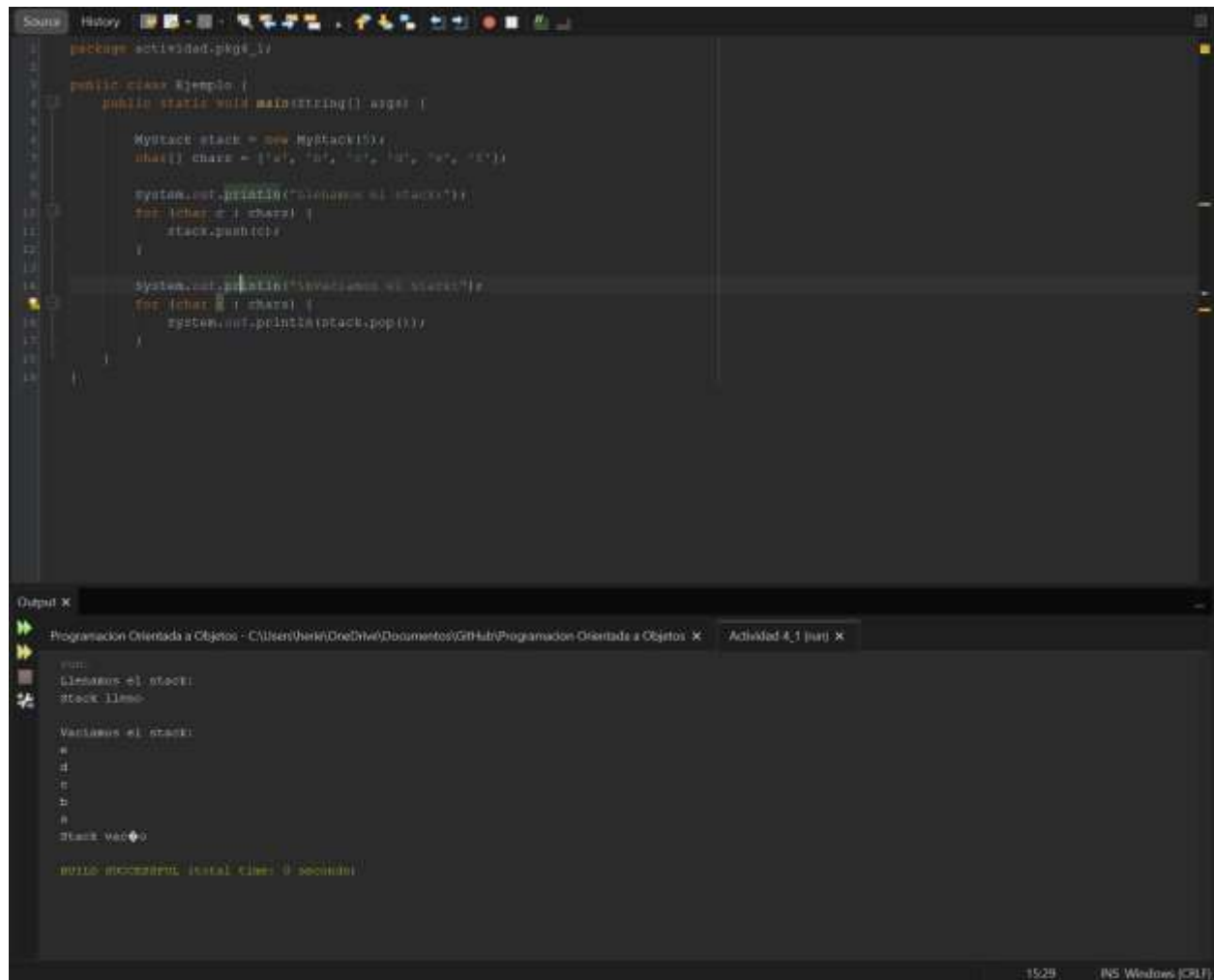
VI24-LIS1022-1: Programación Orientada a Objetos

Dr. Daniel Marcelo Gonzalez Arriaga

30 de mayo de 2024

Actividad 4_1

Stack



The screenshot shows an IDE with a Java source file and its output. The source code defines a `MyStack` class and a `main` method that pushes and pops characters from a stack.

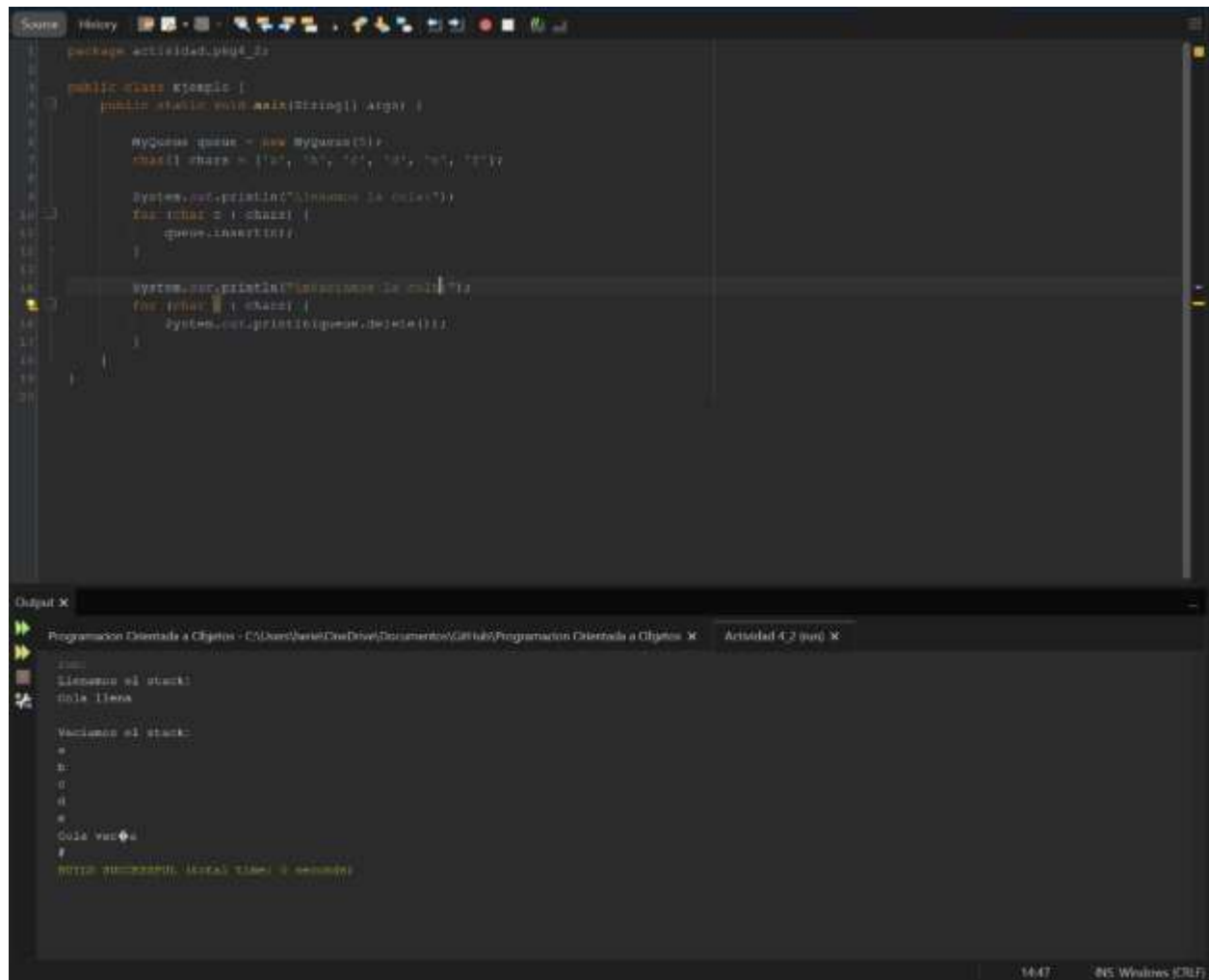
```
1 package actividad.pkg1;  
2  
3 public class Ejemplo {  
4     public static void main(String[] args) {  
5  
6         MyStack stack = new MyStack(5);  
7         char[] chars = {'a', 'b', 'c', 'd', 'e', 'f'};  
8  
9         System.out.println("Elementos en stack:");  
10        for (char c : chars) {  
11            stack.push(c);  
12        }  
13  
14        System.out.println("Variables en stack:");  
15        for (char c : chars) {  
16            System.out.println(stack.pop());  
17        }  
18    }  
19 }
```

The output window shows the execution results:

```
run:  
Elementos en stack:  
Stack lleno  
  
Variables en stack:  
e  
d  
c  
b  
a  
Stack vacio  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Actividad 4_2

Queue



The screenshot shows an IDE with a Java file named `Actividad4_2.java` in the package `actividad4_2`. The code defines a `MyQueue` class with a `main` method. The `main` method creates a `MyQueue` object, adds elements 'a', 'b', 'c', 'd', 'e', and 'f' to the queue, prints the queue's state, and then removes all elements. The output window shows the execution results, including the state of the stack and the final output of the program.

```
package actividad4_2;

public class MyQueue {
    public static void main(String[] args) {

        MyQueue queue = new MyQueue(5);
        char[] chars = {'a', 'b', 'c', 'd', 'e', 'f'};

        System.out.println("Iniciamos la cola");
        for (char c : chars) {
            queue.insert(c);
        }

        System.out.println("Iniciamos la cola");
        for (char c : chars) {
            System.out.println(queue.delete());
        }
    }
}
```

Output

Programacion Orientada a Objetos - C:\Users\user\OneDrive\Documents\Actividad4_2\Programacion Orientada a Objetos - Actividad 4_2 (puro) X

run:

Inicio de la cola

Estado de la cola:

a

b

c

d

e

f

Estado de la cola

f

NOTA: Se ejecutó el programa en 0 segundos

Actividad 4_3

Queue

```
Source rhtoty
package actividad4.prg4_3;

/**
 * Implementa un queue básico, usando la abstracción de Queue de
 * añadir elementos al final de una cola y eliminar elementos de la parte
 * delantera.
 */

import java.util.Queue;
import java.util.LinkedList; // LinkedList es una implementación que utiliza una lista para almacenar los elementos de la cola (Queue).

public class QueueJava {

    public static void main(String[] args) {
        // Crear una cola. LinkedList hace que sea fácil implementar una cola en Java.
        Queue<String> queue = new LinkedList<>(); // Queue, LinkedList

        // Añadir elementos al final de la cola.
        queue.add("E");
        queue.add("una");
        queue.add("cola");

        System.out.println("Mi cola: " + queue);

        // Eliminar elementos de la parte delantera
        String primero = queue.remove();
        System.out.println("Elemento eliminado: " + primero);
        System.out.println("Cola después de eliminar: " + queue);
    }
}
```

Output x

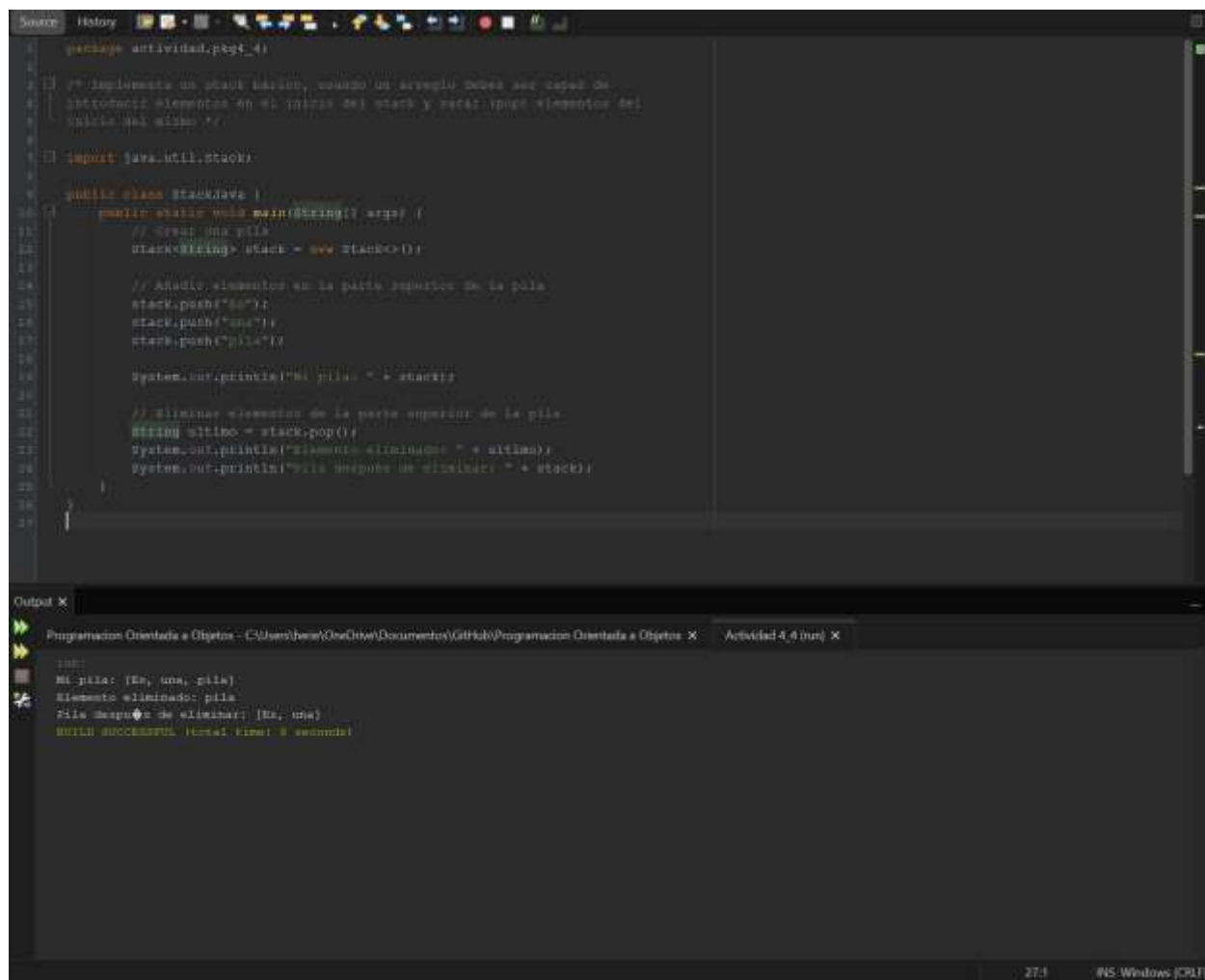
Programacion Orientada a Objetos - C:\Users\javier\OneDrive\Documents\Github\Programacion Orientada a Objetos x Actividad 4_3 (url) x

run:

Mi cola: [E, una, cola]
Elemento eliminado: E
Cola después de eliminar: [una, cola]
SUCCESSFUL (total time: 0 seconds)

1:1 INS Windows (CRLF)

Actividad 4_4



The screenshot displays an IDE with a Java source file and its execution output. The source code implements a stack using an array, with methods for pushing and popping elements. The output window shows the execution results, including the initial state of the stack, the element being removed, and the state after removal.

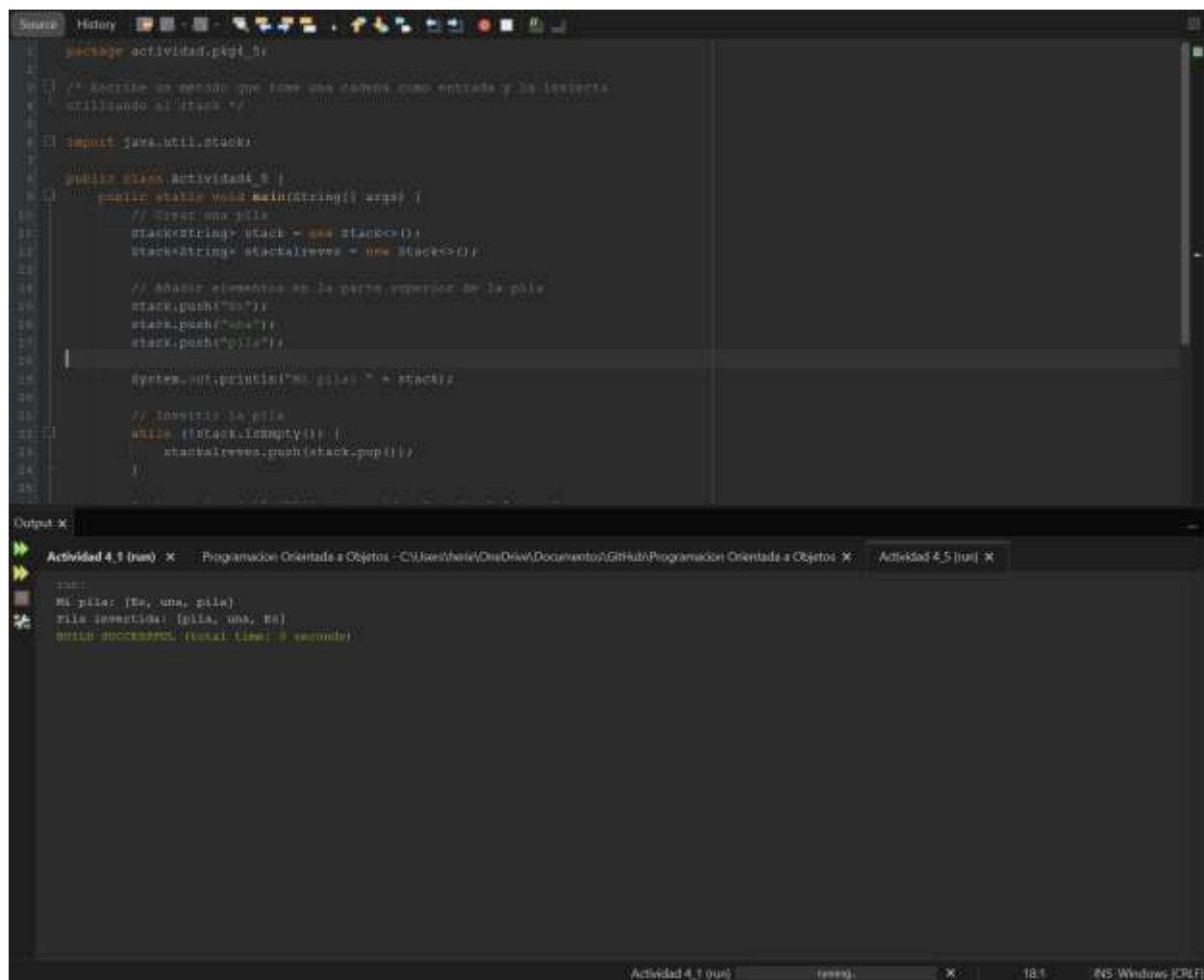
```
1 package actividad.pkg4;
2
3 /** Implementa un stack básico, usando un arreglo donde sea capaz de
4  * introducir elementos en el inicio del stack y sacar (pop) elementos del
5  * inicio del mismo */
6
7 import java.util.Stack;
8
9 public class StackJava {
10     public static void main(String[] args) {
11         // crear una pila
12         Stack<String> stack = new Stack<>();
13
14         // Añadir elementos en la parte superior de la pila
15         stack.push("a");
16         stack.push("b");
17         stack.push("c");
18
19         System.out.println("Mi pila: " + stack);
20
21         // Eliminar elementos de la parte superior de la pila
22         String ultimo = stack.pop();
23         System.out.println("Elemento eliminado: " + ultimo);
24         System.out.println("Pila después de eliminar: " + stack);
25     }
26 }
27
```

Output X

```
Programacion Orientada a Objetos - C:\Users\hewlett\OneDrive\Documents\Github\Programacion Orientada a Objetos x Actividad 4.4 (run) x
run:
Mi pila: [a, b, c]
Elemento eliminado: c
Pila después de eliminar: [a, b]
BUILD SUCCESSFUL (total time: 0 seconds)
```

27.1 MS-Windows [CRLF]

Actividad 4_5



The screenshot shows an IDE with a Java source file and its output. The source code is as follows:

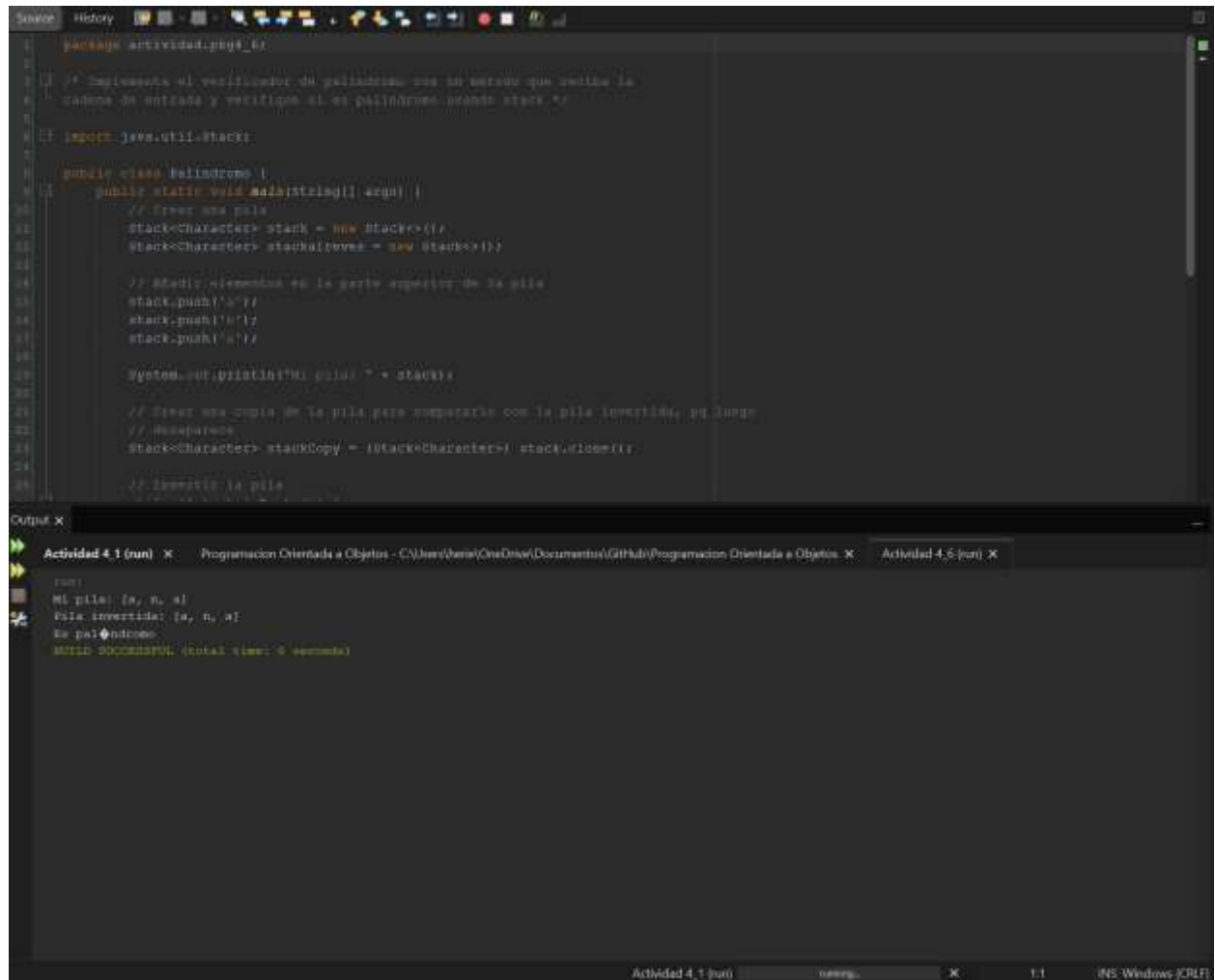
```
1 package actividad.pagi_5;
2
3 /* Escribe un metodo que tome una cadena como entrada y la invierta
4    utilizando el stack */
5
6 import java.util.Stack;
7
8 public class Actividad4_5 {
9     public static void main(String[] args) {
10         // Crear una pila
11         Stack<String> stack = new Stack<>();
12         Stack<String> stackalreves = new Stack<>();
13
14         // Añadir elementos en la parte superior de la pila
15         stack.push("Es");
16         stack.push("una");
17         stack.push("pila");
18
19         System.out.println("Mi pila: " + stack);
20
21         // Invertir la pila
22         while (!stack.isEmpty()) {
23             stackalreves.push(stack.pop());
24         }
25     }
26 }
```

The output window shows the following results:

```
run:
Mi pila: [Es, una, pila]
Pila invertida: [pila, una, Es]
BUILD SUCCESSFUL (total time: 0 segundos)
```

The IDE interface includes a top toolbar, a 'Source' tab, and a bottom status bar showing 'Actividad 4_5 (run)', 'run', '18.1', and 'MS Windows [CRLF]'.

Actividad 4_6



```
1 package actividad.pkg4_6;
2
3 /* Representa el verificador de palindromo con su entrada que recibe la
4 cadena de entrada y verifica si es palindromo usando stack */
5
6 import java.util.Stack;
7
8 public class Palindromo {
9     public static void main(String[] args) {
10         // crear una pila
11         Stack<Character> stack = new Stack<>();
12         Stack<Character> stackInvertida = new Stack<>();
13
14         // Añadir elementos en la parte superior de la pila
15         stack.push('a');
16         stack.push('b');
17         stack.push('c');
18
19         System.out.println("La pila: " + stack);
20
21         // crear una copia de la pila para compararla con la pila invertida, en luego
22         // desapilamos
23         Stack<Character> stackCopy = (Stack<Character>) stack.clone();
24
25         // Invertir la pila
26         while (!stack.isEmpty()) {
27             stackCopy.push(stack.pop());
28         }
29     }
30 }
```

Output x

```
run:
El pila: [a, b, c]
Pila invertida: [c, b, a]
Es palindromo:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Actividad 4.1 (run) x Programacion Orientada a Objetos - C:\Users\user\OneDrive\Documents\Github\Programacion Orientada a Objetos x Actividad 4.5 (run) x

Actividad 4.1 (run) none 1:1 (NS-Window-CRLF)

Actividad 5_1

```

Output X
Actividad 4.1 (run) X Programacion Orientada a Objetos - C:\Users\Daniel\OneDrive\Documentos\GitHub\Programacion Orientada a Objetos. X

1. Llenar y mostrar una matriz.
2. Sumar la diagonal de una matriz.
3. Encontrar el valor máximo en una matriz.
4. Sumar dos matrices.
5. Multiplicar una matriz por un escalar.
6. Multiplicar dos matrices.
7. Transponer una matriz.
8. Salir.

¿Que ejemplo? :

Ingrese el tamaño de la matriz cuadrada:
3

Ingrese el valor de la posición (0)(0): 1
Ingrese el valor de la posición (0)(1): 2
Ingrese el valor de la posición (1)(0): 3
Ingrese el valor de la posición (1)(1): 4

La matriz es:
1 2
3 4

La suma de la diagonal es: 5

1. Llenar y mostrar una matriz.
2. Sumar la diagonal de una matriz.
3. Encontrar el valor máximo en una matriz.
4. Sumar dos matrices.
5. Multiplicar una matriz por un escalar.
6. Multiplicar dos matrices.
7. Transponer una matriz.
8. Salir.

¿Que ejemplo?

```


Actividad 5_2

```

Output X
Actividad 4.1 (run) X Programacion Orientada a Objetos - C:\Users\jheru\OneDrive\Documents\GitHub\Programacion Orientada a Objetos. X

La matriz es:
1  2
3  4

La suma de la diagonal es: 5

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆Que escoges? :

Ingrese el número de columnas: 2

Ingrese el número de filas: 2

Ingrese el valor de la posición (0)(0): 2
Ingrese el valor de la posición (0)(1): 14
Ingrese el valor de la posición (1)(0): 3
Ingrese el valor de la posición (1)(1): 4

La matriz es:
2  14
3  4

El valor máximo de la matriz es: 14

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆Que escoges? :

```

Actividad 4.1 (run)

1/1

R05

Actividad 5_3

```

Output X
Actividad 4.1 (run) X Programacion Orientada a Objetos - C:\Users\halei\OneDrive\Documents\GitHub\Programacion Orientada a Objetos X
1. Encontrar el valor máximo en una matriz
2. Sumar dos matrices
3. Multiplicar una matriz por un escalar
4. Multiplicar dos matrices
5. Transponer una matriz
6. Salir

¿Que escoges? 2

Vamos a sumar dos matrices, el tamaño va a ser el mismo

Ingrese el número de columnas: 3

Ingrese el número de filas: 2

Primera matriz:

Ingrese el valor de la posición [0][0]: 1
Ingrese el valor de la posición [0][1]: 4
Ingrese el valor de la posición [0][2]: 5

Segunda matriz:

Ingrese el valor de la posición [0][0]: 3
Ingrese el valor de la posición [0][1]: 2
Ingrese el valor de la posición [0][2]: 4

suma de matrices:

La matriz es:
3 4 5

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

¿Que escoges?

```

Actividad 5_4

```

Output x
Actividad 4.1 (run) x Programacion Orientada a Objetos - C:\Users\henk\OneDrive\Documentos\Github\Programacion Orientada a Objetos x

main
1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆Que escoges? 1

Ingrese el numero de columnas: 2

Ingrese el numero de filas: 2

Ingrese el valor de la posición [0][0]: 4
Ingrese el valor de la posición [0][1]: 3
Ingrese el valor de la posición [1][0]: 2
Ingrese el valor de la posición [1][1]: 3

Vamos a multiplicar la suma de la matriz por un escalar:
Ingrese el escalar: 2

La matriz es:
20    16
10     6

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆Que escoges?

```

Actividad 5_5

```

Output X
Actividad 4.1 (run) X Programacion Orientada a Objetos - C:\Users\jereh\OneDrive\Documents\GitHub\Programacion Orientada a Objetos X
1. Multiplicar una matriz por un escalar
2. Multiplicar dos matrices
3. Transponer una matriz
4. Salir

Que escogemos? 2

Primera matriz:

Ingrese el numero de columnas: 3

Ingrese el numero de filas: 2

Segunda matriz:

Ingrese el numero de columnas: 3

Ingrese el numero de filas: 2

Primera matriz:

Ingrese el valor de la posición [0][0]: 1
Ingrese el valor de la posición [0][1]: 2
Ingrese el valor de la posición [1][0]: 3
Ingrese el valor de la posición [1][1]: 1

Segunda matriz:

Ingrese el valor de la posición [0][0]: 3
Ingrese el valor de la posición [0][1]: 2
Ingrese el valor de la posición [1][0]: 2
Ingrese el valor de la posición [1][1]: 1

Multiplicación de matrices:

La matriz es:
7 4
11 7

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

Actividad 4.1 (run) 25445 85

```

$$\begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix}$$

Solución

$$\begin{pmatrix} 7 & 4 \\ 11 & 7 \end{pmatrix}$$

Actividad 5_6

```

Output X
Actividad 4.1 (run) X Programacion Orientada a Objetos - C:\Users\Javier\OneDrive\Documents\GitHub\Programacion Orientada a Objetos X
Multiplicación de matrices

La matriz es:
J 4
II 1

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆ Que escoges?

Ingresa el numero de columnas: 3

Ingresa el numero de filas: 3

Ingresa el valor de la posición (0)(0): 1
Ingresa el valor de la posición (0)(1): 2
Ingresa el valor de la posición (0)(2): 3
Ingresa el valor de la posición (1)(0): 4
Ingresa el valor de la posición (1)(1): 5
Ingresa el valor de la posición (1)(2): 6
Ingresa el valor de la posición (2)(0): 7
Ingresa el valor de la posición (2)(1): 8
Ingresa el valor de la posición (2)(2): 9

Transposición de la matriz:

La matriz es:
1 4 7
2 5 8
3 6 9

1. Llenar y mostrar una matriz
2. Sumar la diagonal de una matriz
3. Encontrar el valor máximo en una matriz
4. Sumar dos matrices
5. Multiplicar una matriz por un escalar
6. Multiplicar dos matrices
7. Transponer una matriz
8. Salir

◆ Que escoges?

```