Ingeniería inversa del archivo JAR ofuscado

¿Qué tanto se dificulta la lectura?

El JAR ofuscado se vuelve más difícil de leer ya que los nombres que tienen las clases, métodos y variables cambian por identificaciones que no tienen nada que ver con la sintaxis del código lo que evita que se pueda reconocer la funcionalidad del código.

¿Se pierde claridad estructural?

Si se pierde la claridad del código tanto a nivel lógico como la sintaxis. El código mantiene su estructura, pero la ofuscación elimina la información importa lo que hace que el código sea mucho más difícil de comprender el código.

¿Sigue siendo posible entender la lógica?

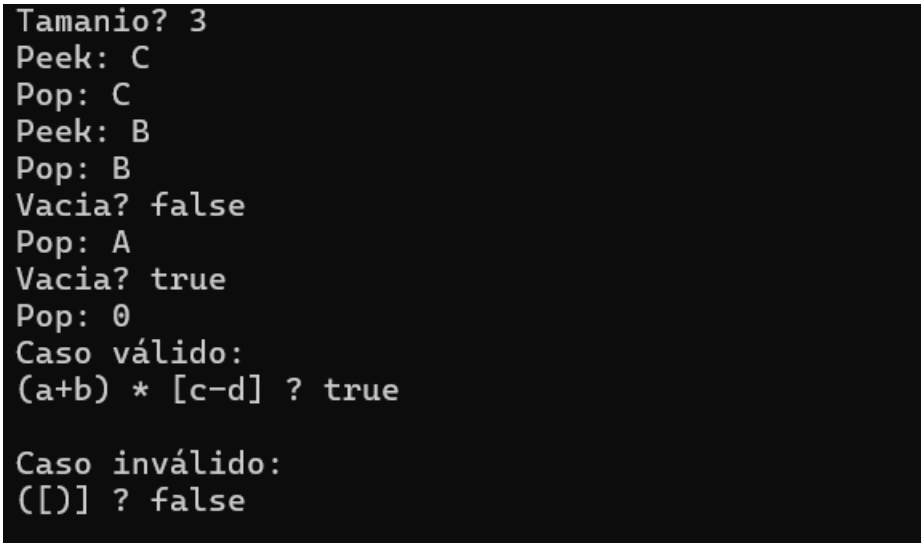
Es posible seguir entendiendo la lógica del programa, pero se necesita mucho mas tiempo y experiencia para lograr entender la lógica del código, la ofuscación cumple con el objetivo de dificultar la ingeniería inversa del código.

Evidencias – Prueba de regresión desde consola

Comando ejecutado

```
C:\Users\Asus\Desktop\Practica Pilas + Maven\stackHandler\target>java -cp "stackHandler-0.0.1-SNAPSHOT-obf.jar;libs/*" s
tackHandler.handler.Main "(a+b)"
```

Salida en consola



Comparación antes y después de la ofuscación

Aspecto	JAR sin ofuscar	JAR ofuscado
Ejecución desde consola	Funciona correctamente	Funciona correctamente
Resultado lógico	Correcto	Correcto
Nombres de clases y métodos	Legibles y descriptivos	Ofuscados y no descriptivos
Comprensión del código	Alta	Baja
Comportamiento del programa	Correcto	Correcto

Conclusión

La ofuscación no altera el comportamiento funcional del programa.
El JAR ofuscado produce exactamente la misma salida que el JAR original al ejecutarse desde consola, confirmando que la lógica del programa se mantiene intacta.
La ofuscación cumple su objetivo al dificultar la ingeniería inversa sin afectar la ejecución ni los resultados.