



Instituto Universitario de Yucatán

“procesos y subprocesos

Ingeniería en sistemas computacionales y diseño de software, Instituto Universitario de
Yucatán

Licenciada por la Alejandra Landero Heredia

Fecha: 10/08/2025

```
BankAccount.java
38 }
39
40 public static void main(String[] args){
41     BankAccount account = new BankAccount();
42
43     Thread depositThread1 = new Thread() -> account
44     .deposit(1000));
45     Thread depositThread2 = new Thread() -> account
46     .deposit(300));
47     Thread withdrawalThread1 = new Thread() ->
48     account.withdraw(150));
49     Thread withdrawalThread2 = new Thread() ->
50     account.withdraw(1200));
51
52     depositThread1.start();
53     depositThread2.start();
54     withdrawalThread1.start();
}
```

Output

```
Deposito: 1000.0
Saldo despues del deposito: 1000.0
Deposito: 300.0
Saldo despues del deposito: 1300.0
Retiro: 150.0
Saldo despues del retiro:1150.0
Intento de retiro:1200.0
Saldo insuficiente, retiro cancelado.

=== Code Execution Successful ===
```

Online Java Compiler

```
BankAccount.java
17     System.out.println("Deposito: " + amount);
18     System.out.println("Saldo despues del deposito: " +
19         balance);
20 } finally {
21     lock.unlock();
22 }
23 public void withdraw(double amount){
24     lock.lock();
25     try {
26         if(balance >= amount){
27             balance -= amount;
28             System.out.println("Retiro: " + amount);
29             System.out.println("Saldo despues del retiro:" + balance
30             );
31         } else {
32             System.out.println("Intento de retiro:" + amount);
33             System.out.println("Saldo insuficiente, retiro
34             cancelado.");
35         }
36     } finally {
37         lock.unlock();
38     }
39 }
40 public static void main(String[] args){
41     BankAccount account = new BankAccount();
42
43     Thread depositThread1 = new Thread() -> account
44     .deposit(1000));
45     Thread depositThread2 = new Thread() -> account
46     .deposit(300));
47     Thread withdrawalThread1 = new Thread() ->
48     account.withdraw(150));
49     Thread withdrawalThread2 = new Thread() ->
50     account.withdraw(1200));
51
52     depositThread1.start();
53     depositThread2.start();
54     withdrawalThread1.start();
}
```

Output

```
Deposito: 1000.0
Saldo despues del deposito: 1000.0
Deposito: 300.0
Saldo despues del deposito: 1300.0
Retiro: 150.0
Saldo despues del retiro:1150.0
Intento de retiro:1200.0
Saldo insuficiente, retiro cancelado.

=== Code Execution Successful ===
```

```
BankAccount.java
34 }
35 } finally {
36     lock.unlock();
37 }
38 }
39
40 public static void main(String[] args){
41     BankAccount account = new BankAccount();
42
43     Thread depositThread1 = new Thread() -> account
44     .deposit(1000));
45     Thread depositThread2 = new Thread() -> account
46     .deposit(300));
47     Thread withdrawalThread1 = new Thread() ->
48     account.withdraw(150));
49     Thread withdrawalThread2 = new Thread() ->
50     account.withdraw(1200));
51
52     depositThread1.start();
53     depositThread2.start();
54     withdrawalThread1.start();
}
```

Output

```
Deposito: 1000.0
Saldo despues del deposito: 1000.0
Deposito: 300.0
Saldo despues del deposito: 1300.0
Retiro: 150.0
Saldo despues del retiro:1150.0
Intento de retiro:1200.0
Saldo insuficiente, retiro cancelado.

=== Code Execution Successful ===
```