

Contenido

Exercise 1: Using the Java API Documentation.....	2
Exercise 2: Exploring Encapsulation	2
Task 1 – Deleting the Account Class	2
Task 2 – Creating the Account Class	3
Task 3 – Creating the TestAccount2 Class.....	3
Task 4 – Compiling the TestAccount2 Class	4
Task 5 – Running the TestAccount2 Program	4
Exercise 3: Exploring Encapsulation	4
Task 1 – Modifying the Account Class	4
Task 2 – Modifying the TestAccount Class.....	5
Task 3 – Compiling the TestAccount Class	6
Task 4 – Running the TestAccount Program	6
Exercise 4: Creating Java Packages	7
Task 1 – Creating the Java Packages	7
Task 2 – Moving and Modifying the Account Class	7
Task 3 – Moving the TestAccount Class	8
Task 4 – Compiling the TestAccount Class	8
Task 5 – Running the TestAccount Program	8

Exercise 1: Using the Java API Documentation

The screenshot shows the Java API documentation for the `java.text.Format` class. On the left is a navigation pane with a tree view of the Java class hierarchy, including packages like `java.security`, `java.time`, and `java.text`. The `Format` class is selected. The main pane displays the `format` method signature: `public StringBuffer format(Object number, StringBuffer toAppendTo, FieldPosition pos)`. Below the signature is a description: "Formats a number and appends the resulting text to the given string buffer. The number can be of any subclass of `Number`." This is followed by a detailed implementation note: "This implementation extracts the number's value using `Number.longValue()` for all integral type values that can be converted to `long` without loss of information, including `BigInteger` values with a bit length of less than 64, and `Number.doubleValue()` for all other types. It then calls `format(Long, java.lang.StringBuffer, java.text.FieldPosition)` or `format(double, java.lang.StringBuffer, java.text.FieldPosition)`. This may result in loss of magnitude information and precision for `BigInteger` and `BigDecimal` values." Further down, it lists "Specified by:" (format in class `Format`), "Parameters:" (number, toAppendTo, pos), "Returns:" (the value passed in as toAppendTo), and "Throws:" (`IllegalArgumentException`, `NullPointerException`, `ArithmeticException`). A "See Also:" section points to `FieldPosition`. Below the `format` method is the `parseObject` method signature: `public final Object parseObject(String source, ParsePosition pos)`.

Exercise 2: Exploring Encapsulation

Task 1 – Deleting the Account Class

The screenshot shows an IDE interface with a dark theme. On the left, a file explorer shows a project named "Lab2" containing two files: "Account.java" and "TestAccount.java". A right-click context menu is open over "Account.java". The menu items include "File History", "Open Timeline", "Cut" (with shortcut Ctrl+X), "Copy" (with shortcut Ctrl+C), "Copy Path" (with shortcut Shift+Alt+C), "Copy Relative Path" (with shortcut Ctrl+K Ctrl+Shift+C), "Copy Remote File URL", "Copy Remote File URL From...", "Rename..." (with shortcut F2), "Delete" (highlighted in blue), and "Run Java". The "Delete" option has a secondary "Delete" label next to it. On the right side of the IDE, a code editor shows the implementation of the `Account` class, including methods like `getBalance()`, `deposit(double amt)`, and `withdraw(double amt)`.

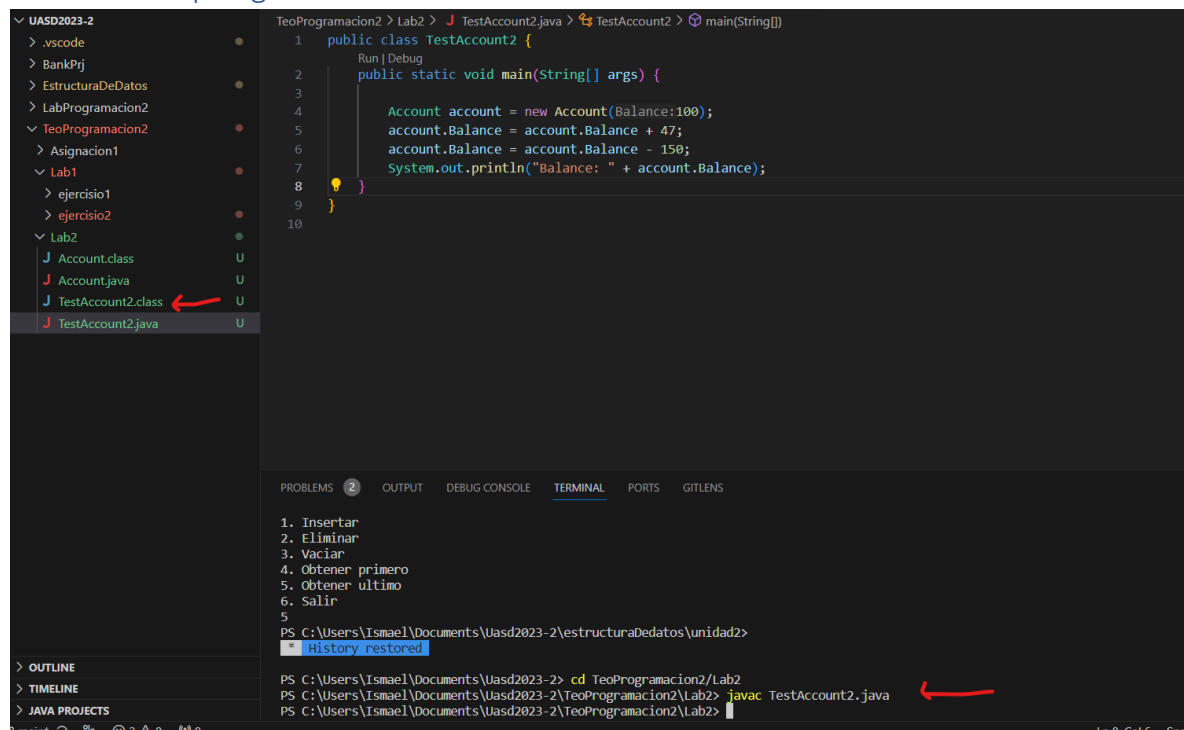
Task 2 – Creating the Account Class

```
TeoProgramacion2 > Lab2 > Account.java > Account > Account(double)
1 public class Account {
2
3     public double Balance;
4
5     public Account( double Balance) {
6         this.Balance = Balance;
7     }
8 }
9
```

Task 3 – Creating the TestAccount2 Class

```
TeoProgramacion2 > Lab2 > TestAccount2.java > TestAccount2 > main(String[])
1 public class TestAccount2 {
2     Run | Debug
3     public static void main(String[] args) {
4         Account account = new Account(Balance:100);
5         account.Balance = account.Balance + 47;
6         account.Balance = account.Balance - 150;
7         System.out.println("Balance: " + account.Balance);
8     }
9 }
10
```

Task 4 – Compiling the TestAccount2 Class

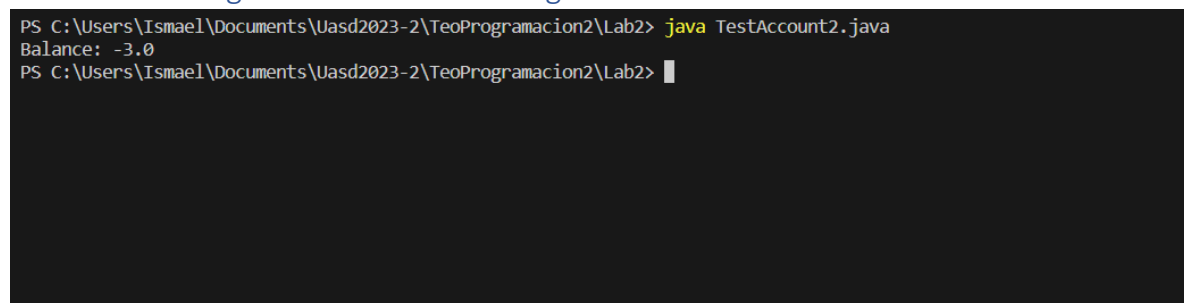


```
TeoProgramacion2 > Lab2 > J TestAccount2.java > TestAccount2 > main(String[])
1 public class TestAccount2 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         Account account = new Account(Balance:100);
6         account.Balance = account.Balance + 47;
7         account.Balance = account.Balance - 150;
8         System.out.println("Balance: " + account.Balance);
9     }
10 }
```

1. Insertar
2. Eliminar
3. Vaciar
4. Obtener primero
5. Obtener ultimo
6. Salir
5
PS C:\Users\Ismael\Documents\Uasd2023-2\estructuraDeDatos\unidad2>
* History restored

PS C:\Users\Ismael\Documents\Uasd2023-2> cd TeoProgramacion2\Lab2
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2> javac TestAccount2.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2>

Task 5 – Running the TestAccount2 Program



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2> java TestAccount2.java
Balance: -3.0
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2>
```

Exercise 3: Exploring Encapsulation

Task 1 – Modifying the Account Class



```
public class Account {

    private double Balance;

    public Account(double Balance) {
        this.Balance = Balance;
    }

    public void deposit(double amt) {
```

```
        Balance = Balance + amt;
    }

    public void withdraw(double amt) {
        if (amt <= Balance) {
            Balance = Balance - amt;
        }
    }

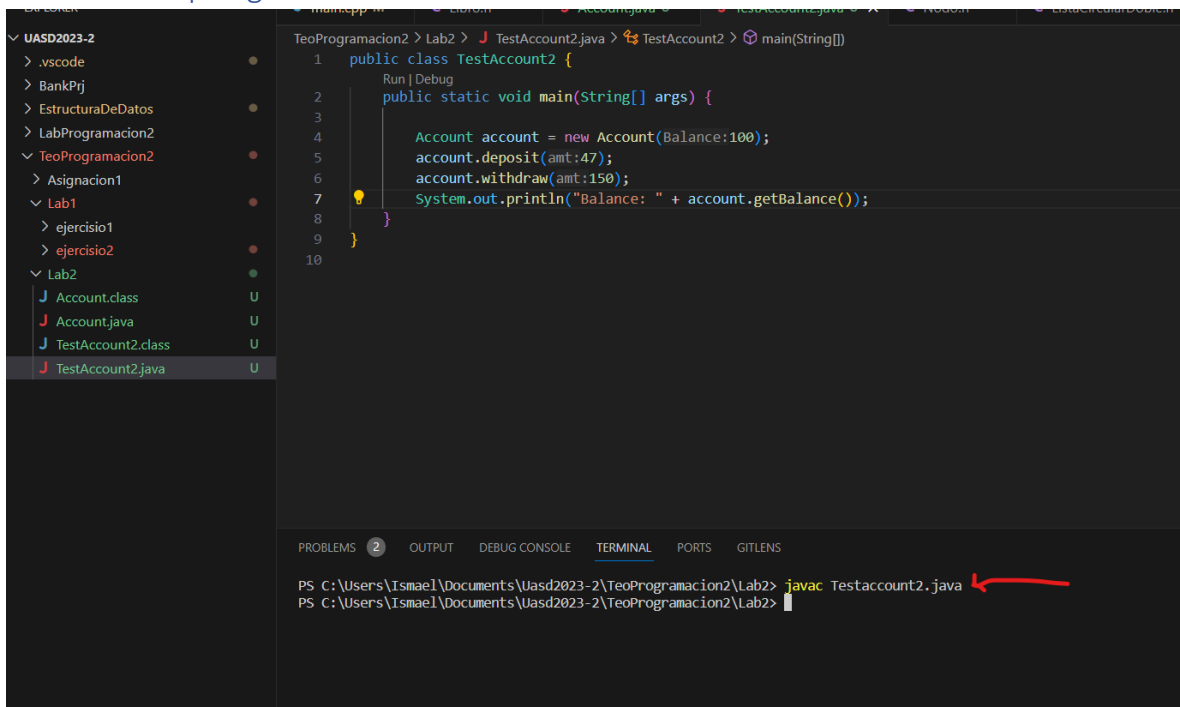
    public double getBalance() {
        return Balance;
    }
}
```

Task 2 – Modifying the TestAccount Class

```
public class TestAccount2 {
    public static void main(String[] args) {

        Account account = new Account(100);
        account.deposit(47);
        account.withdraw(150);
        System.out.println("Balance: " + account.getBalance());
    }
}
```

Task 3 – Compiling the TestAccount Class

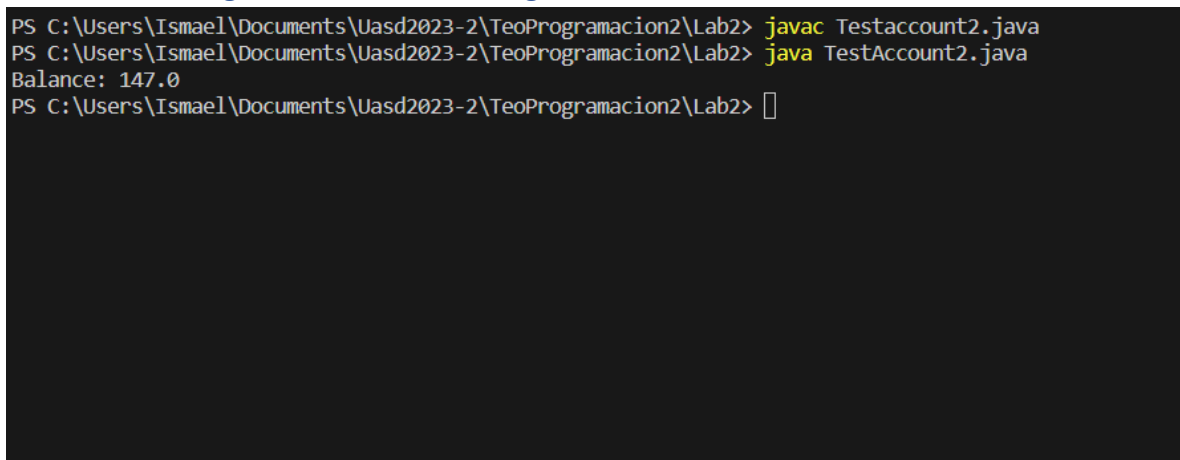


```
TeoProgramacion2 > Lab2 > J TestAccount2.java > TestAccount2 > main(String[])
1 public class TestAccount2 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         Account account = new Account(Balance:100);
6         account.deposit(amt:47);
7         account.withdraw(amt:150);
8         System.out.println("Balance: " + account.getBalance());
9     }
10 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2> javac Testaccount2.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2>
```

Task 4 – Running the TestAccount Program



```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2> javac Testaccount2.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2> java TestAccount2.java
Balance: 147.0
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\Lab2>
```

Exercise 4: Creating Java Packages

Task 1 – Creating the Java Packages

```
El número de serie del volumen es D887-56B3
C:.\
├── com
│   └── mybank
│       ├── domain
│       └── test
```

Task 2 – Moving and Modifying the Account Class

```
package com.mybank.domain;

public class Account {

    private double Balance;

    public Account(double Balance) {
        this.Balance = Balance;
    }

    public void deposit(double amt) {
        Balance = Balance + amt;
    }

    public void withdraw(double amt) {
        if (amt <= Balance) {
            Balance = Balance - amt;
        }
    }

    public double getBalance() {
        return Balance;
    }
}
```

Task 3 – Moving the TestAccount Class

```
package com.mybank.test;

import com.mybank.domain.Account;

public class TestAccount2 {
    public static void main(String[] args) {

        Account account = new Account(100);
        account.deposit(47);
        account.withdraw(150);
        System.out.println("Balance: " + account.getBalance());
    }
}
```

Task 4 – Compiling the TestAccount Class

```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\lab2> javac -cp . com/mybank/test/TestAccount2.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\lab2> █
```

Task 5 – Running the TestAccount Program

```
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\lab2> javac -cp . com/mybank/test/TestAccount2.java
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\lab2> java -cp . com/mybank/test/TestAccount2.java
Balance: 147.0
PS C:\Users\Ismael\Documents\Uasd2023-2\TeoProgramacion2\lab2> █
```