Nombres: Catalina Quidel -- Natalia Perez
Curso: Programación Orientada a Objetos

Objetivo: aplicar los conceptos de clases e instanciación de objetos con LDP Java, usando el ocultamiento de datos.

Preguntas a responder:

Pgta 1. ¿Desde dónde es posible acceder a la variable time?

1. Cualquier otra clase
2. java.util package
3. myUtilities package
4. La clase TodaysDate
5. La clase TodaysDate y sus sub clase

Pgta 2. ¿Quiénes pueden acceder al atributo day?

day es accesible desde cualquier otra clase en cualquier paquete. Osea está disponible para todos.

Pgta 3. ¿Qué atributos de la clase tienen el método de acceso más restrictivo?

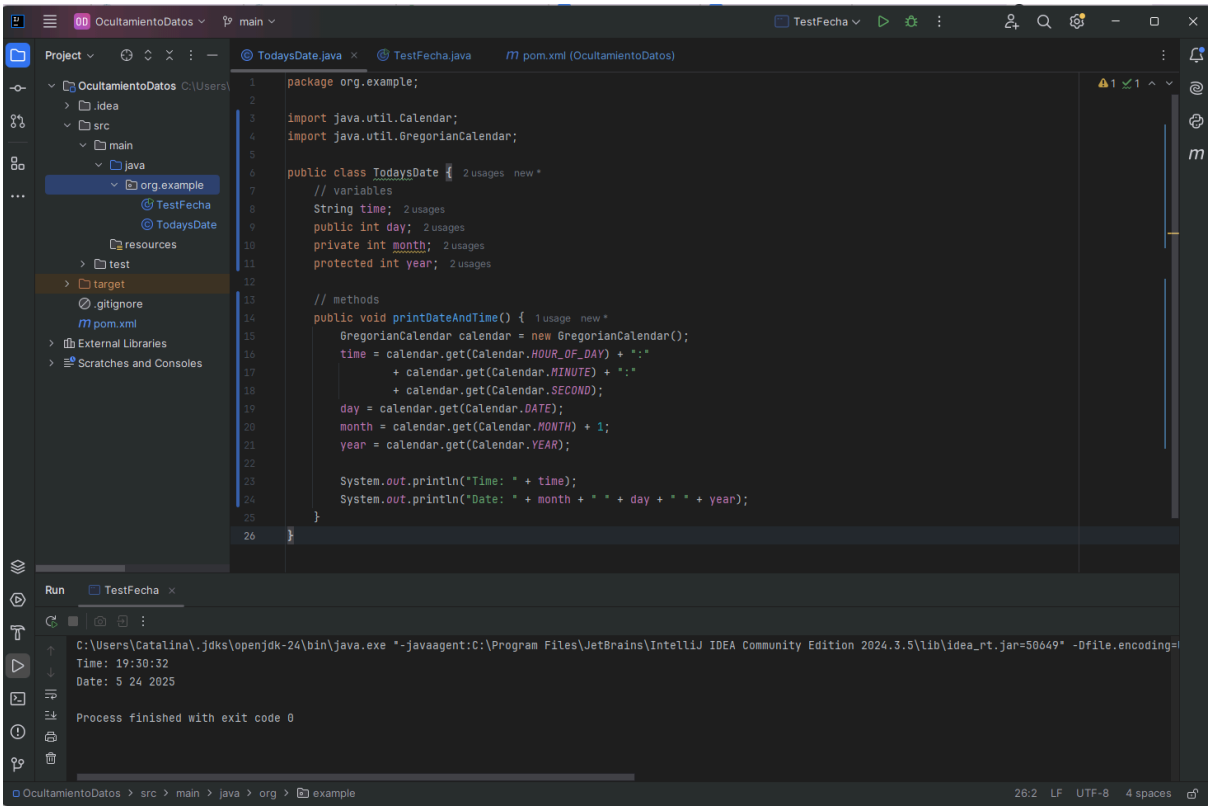
1. day
2. month
3. time
4. year

Pgta 4. ¿Desde dónde se puede acceder al atributo year?

1. Sólo dentro del package myUtilities.
2. Sub clases de TodaysDate en cualquier package
3. La clase TodaysDate
4. Sub clases de TodaysDate

Nombres: Catalina Quidel -- Natalia Perez
Curso: Programación Orientada a Objetos

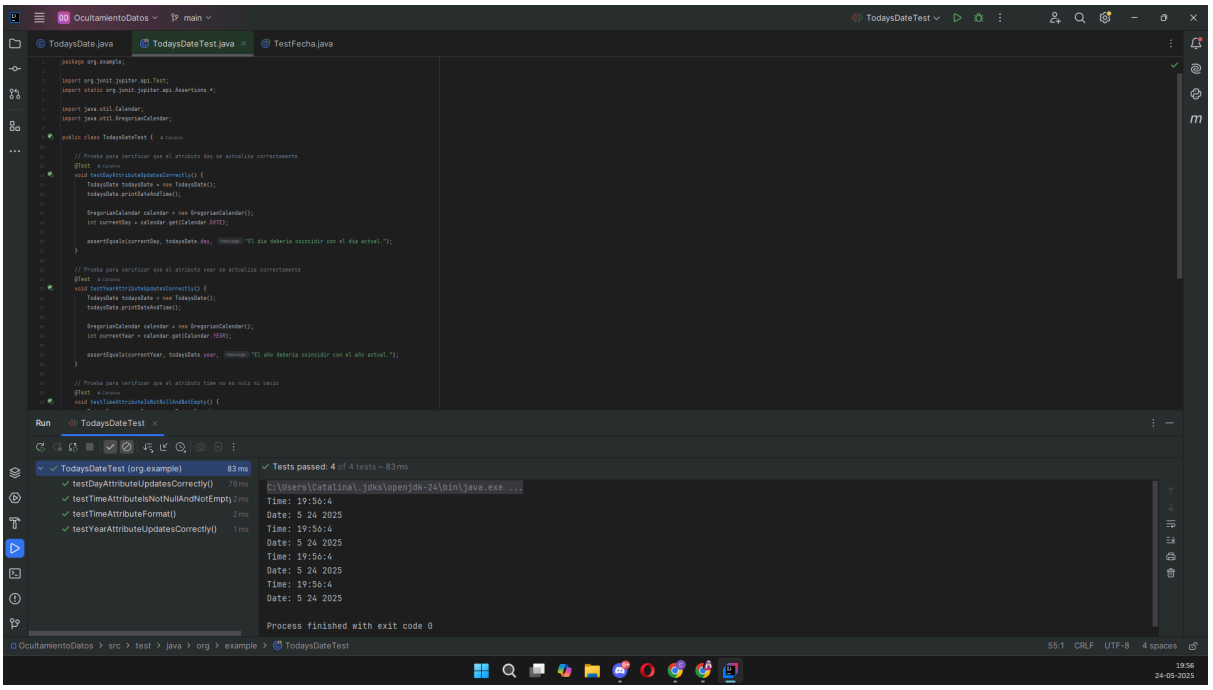
Clase TodaysDate por Catalina Quidel y estructura de paquetes



Compilación exitosa en con la clase pública TestFecha por Catalina Quidel

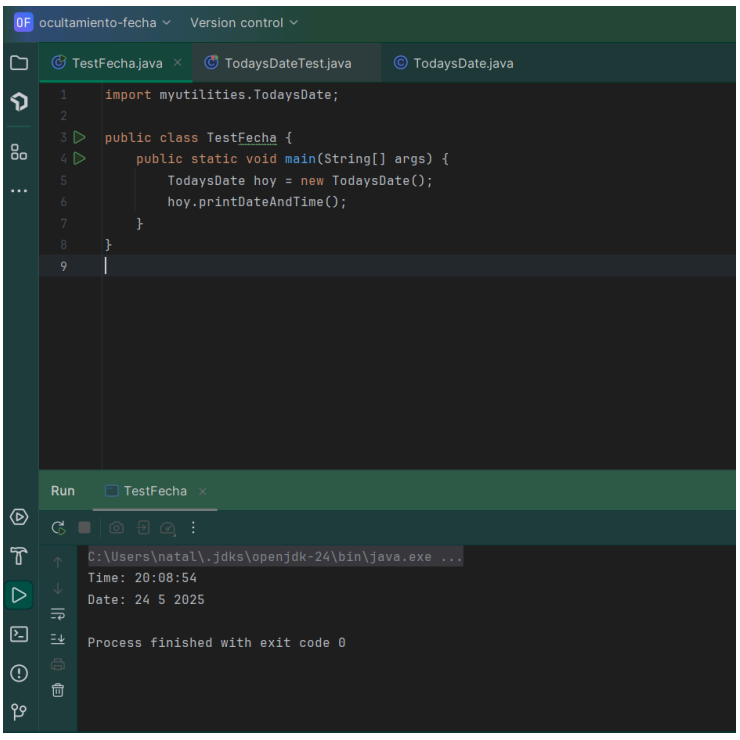
Nombres: Catalina Quidel -- Natalia Perez
Curso: Programación Orientada a Objetos

Pruebas unitarias por Catalina Quidel



El ocultamiento de datos y cómo afecta a las pruebas

Cómo month es una variable private, no se puede acceder directamente desde fuera de la clase TodaysDate, incluyendo la clase de prueba unitaria. Una forma para hacer un Test sobre month sería usar el valor del atributo a través de un método Getter público. Getter que realizó Natalia Perez a continuación.

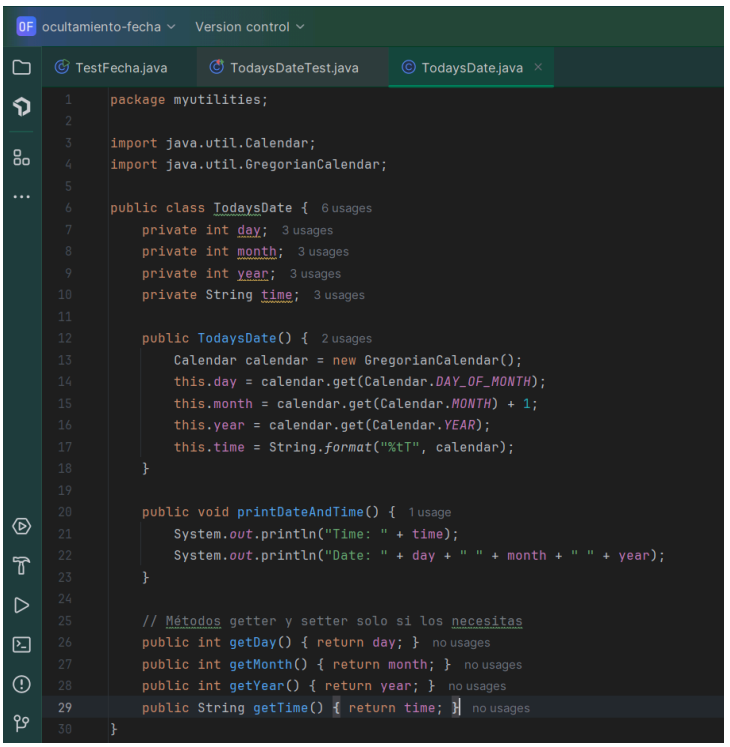


```
import myutilities.TodaysDate;

public class TestFecha {
    public static void main(String[] args) {
        TodaysDate hoy = new TodaysDate();
        hoy.printDateAndTime();
    }
}
```

```
C:\Users\natal\.jdks\openjdk-24\bin\java.exe ...
Time: 20:08:54
Date: 24 5 2025

Process finished with exit code 0
```



```
package myutilities;

import java.util.Calendar;
import java.util.GregorianCalendar;

public class TodaysDate {
    private int day;
    private int month;
    private int year;
    private String time;

    public TodaysDate() {
        Calendar calendar = new GregorianCalendar();
        this.day = calendar.get(Calendar.DAY_OF_MONTH);
        this.month = calendar.get(Calendar.MONTH) + 1;
        this.year = calendar.get(Calendar.YEAR);
        this.time = String.format("%tT", calendar);
    }

    public void printDateAndTime() {
        System.out.println("Time: " + time);
        System.out.println("Date: " + day + " " + month + " " + year);
    }

    // Métodos getter y setter solo si los necesitas
    public int getDay() { return day; }
    public int getMonth() { return month; }
    public int getYear() { return year; }
    public String getTime() { return time; }
}
```

Nombres: Catalina Quidel -- Natalia Perez
Curso: Programación Orientada a Objetos

TestFecha.java | TodaysDateTest.java | TodaysDate.java

```java
import myutilities.TodaysDate;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class TodaysDateTest {

    @Test
    void testObjectNotNull() {
        TodaysDate date = new TodaysDate();
        assertNotNull(date, message: "El objeto TodaysDate no debería ser nulo");
    }

    @Test
    void testDayIsValid() {
        TodaysDate date = new TodaysDate();
        assertTrue( condition: date.getDay() >= 1 && date.getDay() <= 31, message: "El día debe estar entre 1 y 31");
    }

    @Test
    void testMonthIsValid() {
        TodaysDate date = new TodaysDate();
        assertTrue( condition: date.getMonth() >= 1 && date.getMonth() <= 12, message: "El mes debe estar entre 1 y 12");
    }

    @Test
    void testYearIsCurrent() {
        TodaysDate date = new TodaysDate();
        int currentYear = java.util.Calendar.getInstance().get(java.util.Calendar.YEAR);
        assertEquals(currentYear, date.getYear(), message: "El año debe ser el actual");
    }

    @Test
    void testTimeFormat() {
        TodaysDate date = new TodaysDate();
        assertTrue(date.getTime().matches( regex: "\\d{2}:\\d{2}:\\d{2}"), message: "El formato de hora debe ser HH:MM:SS");
    }
}
```

Run | TodaysDateTest

TodaysDateTest 146 ms
- testTimeFormat() 131 ms
- testYearIsCurrent() 7 ms
- testObjectNotNull() 4 ms
- testMonthIsValid() 2 ms
- testDayIsValid() 2 ms

Tests passed: 5 of 5 tests – 146 ms

```
C:\Users\natal\.jdks\openjdk-24\bin\java.exe ...

Process finished with exit code 0
```