```java
import org.openqa.selenium.*;

import org.openqa.selenium.chrome.*;

import org.openqa.selenium.support.ui.*;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.time.temporal.ChronoUnit;

import java.util.List;

import java.util.regex.*;


public class NebulaTest {

    public static void main(String[] args) {

        String searchKey = System.getProperty("search", "paracetamol");


        ChromeOptions options = new ChromeOptions();

        options.addArguments("--incognito");

        WebDriver driver = new ChromeDriver(options);

        WebDriverWait wait = new WebDriverWait(driver, 20);


        try {

            driver.get("https://patinformed.wipo.int/");

            handlePopups(driver);


            WebElement searchBox = wait.until(ExpectedConditions.elementToBeClickable(By.id("home-search-input")));

            searchBox.sendKeys(searchKey + Keys.ENTER);


            wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".results-table tbody tr")));

            WebElement firstResult = driver.findElement(By.cssSelector(".results-table tbody tr td:nth-child(2)"));

            firstResult.click();
```

```java
                wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(".mat-tab-body-
content")));


            List<WebElement> tables = driver.findElements(By.tagName("table"));

            for (WebElement table : tables) {

                List<WebElement> rows = table.findElements(By.tagName("tr"));

                String filing = null, publication = null, grant = null;


                for (WebElement row : rows) {

                    String text = row.getText().toLowerCase();

                    if (text.contains("filing date")) filing = extractDate(row.getText());

                    if (text.contains("publication date")) publication = extractDate(row.getText());

                    if (text.contains("grant date")) grant = extractDate(row.getText());

                }


                if (filing != null || publication != null || grant != null) {

                    printDifferences(filing, publication, grant);

                    break;

                }

            }


        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        } finally {

            driver.quit();

        }

    }


    private static void handlePopups(WebDriver driver) {

        try {

            WebDriverWait wait = new WebDriverWait(driver, 5);
```

```java
        WebElement acceptBtn =
wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector("button.cookie-consent-
accept")));

        acceptBtn.click();

    } catch (Exception ignored) {}

  }


  private static String extractDate(String text) {

    Pattern pattern = Pattern.compile("\\d{4}-\\d{2}-\\d{2}");

    Matcher matcher = pattern.matcher(text);

    return matcher.find() ? matcher.group() : null;

  }


  private static void printDifferences(String filing, String publication, String grant) {

    DateTimeFormatter fmt = DateTimeFormatter.ofPattern("yyyy-MM-dd");


    if (publication != null) System.out.println("Publication date: " + publication);

    if (grant != null) System.out.println("Grant date: " + grant);

    if (filing != null) System.out.println("Filing date: " + filing);


    if (publication != null && grant != null)

      System.out.println("→ Publication ↔ Grant: " +
ChronoUnit.DAYS.between(LocalDate.parse(publication, fmt), LocalDate.parse(grant, fmt)) + "
days");


    if (publication != null && filing != null)

      System.out.println("→ Publication ↔ Filing: " +
ChronoUnit.DAYS.between(LocalDate.parse(publication, fmt), LocalDate.parse(filing, fmt)) + " days");

  if (grant != null && filing != null)

      System.out.println("→ Grant ↔ Filing: " + ChronoUnit.DAYS.between(LocalDate.parse(grant,
fmt), LocalDate.parse(filing, fmt)) + " days");

 }

}
```