**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 2

**З дисципліни:** *“Кросплатформне програмування”*

**На тему:** *“Робота з текстовою інформацією”*

**Лектор:**

доц. каф. ПЗ

Дяконюк Л. М.

**Виконав:**

ст. гр. ПЗ-35

Хруставчук М.Л.

**Прийняв:**

ст. викл. каф. ПЗ

Шкраб Р. Р.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2024 р.

∑= \_\_\_\_\_ .

Львів – 2024

**Тема роботи:** Робота з текстовою інформацією.

**Мета роботи:** Навчитися використовувати основний інструментарій мови для пошуку, зміни, форматування текстової інформації, зокрема здобути навички роботи з датами та часовими змінними в різних часових зонах.

**TЕОРЕТИЧНІ ВІДОМОСТІ**

Важливе місце в обробці інформації займає робота з текстами. Текстові рядки в мові Java представляють екземпляри класу String або класу StringBuffer.

Класи String, StringBuffer, StringBuilder реалізують інтерфейс CharSequence.

На відміну від String, об'єкти типу StringBuffer і StringBuilder можуть бути змінені, вставляючи і додаючи рядки і символи. В багатьох випадках, коли треба змінити довжину рядка типу String, компілятор Java неявно перетворює його до типу StringBuffer, змінює довжину, потім перетворює назад в тип String.

Робота з датою і часом в Java

Новий API java.time складається з 5 пакетів:

1. Java.time – базовий пакет, що містить об'єкти-значення
2. Java.time.chrono – надає доступ до різних календарних систем
3. Java.time.format – дозволяє форматувати та аналізувати дату та час
4. Java.time.temporal – низькорівневий фреймворк та розширені можливості
5. Java.time.zone – підтримує класи для часових поясів

Клас LocalDate є одним з найважливіших у новому API. Це незмінний тип значення, який представляє дату. У ньому немає представлення часу доби або часового поясу. Об'єкти цього класу не змінюються після створення.

Java.time.LocalTime

Це час без дати та часового поясу. Застосовується стандартне визначення часу: час усередині доби, у 24-годинному форматі, починаючи з опівночі.

LocalTime зберігає години, хвилини, секунди і наносекунди. Цей тип підтримує точність аж до наносекунд, фактична точність залежить від реалізації JVM/JDK.

**ЗАВДАННЯ**

*Варіант 3.*

Розробіть програму для обчислення найближчого робочого дня доставки замовлення в онлайн-магазині. Користувач вводить дату замовлення, і програма повинна враховувати робочі дні, робочий графік специфічний для кожного продавця та можливість впливу різних факторів, таких як середній час доставки, віддаленість від магазину тощо.

Додаткові умови:

1. Кожен продавець може мати власний робочий графік, включаючи робочі та неробочі дні.
2. Обчисліть середній час доставки для кожного продавця.
3. Враховуйте часові зони для різних місць доставки та магазину.
4. Розгляньте можливість різних методів доставки з різними строками (наприклад, експрес-доставка, звичайна доставка).
5. Враховуйте можливість виникнення святкових днів та інших відмінностей у графіках доставки.

Пароль повинен містити щонайменше одну велику літеру, одну малу літеру, одну цифру та один спеціальний символ (наприклад: !@#$%^&\*). Довжина повинна бути не менше 8 символів і не більше 20 символів. Знайти можливі слова, кандидати на паролі.

**ХІД ВИКОНАННЯ**

**1. Код розробленої програми**

Main.java

package org.example;  
  
import java.io.\*;  
import java.nio.file.\*;  
import java.util.regex.\*;  
import java.util.\*;  
import java.time.\*;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 char answer;  
 System.*out*.print("Введіть номер завдання, яке треба виконати (1/2): ");  
 answer = scanner.next().charAt(0);  
  
 if (answer == '1') {  
 Seller seller1 = new Seller("Shop1", "Germany", Arrays.*asList*(DayOfWeek.*MONDAY*, DayOfWeek.*TUESDAY*, DayOfWeek.*WEDNESDAY*, DayOfWeek.*THURSDAY*, DayOfWeek.*FRIDAY*),  
 3, 2, ZoneId.*of*("Europe/Berlin"));  
  
 Seller seller2 = new Seller("Shop2", "USA", Arrays.*asList*( DayOfWeek.*TUESDAY*, DayOfWeek.*THURSDAY*, DayOfWeek.*SATURDAY*, DayOfWeek.*SUNDAY*),  
 14, 10, ZoneId.*of*("America/Chicago"));  
  
 HolidayCalendar holidayCalendar = new HolidayCalendar(Arrays.*asList*(LocalDate.*of*(2024, 1, 1), LocalDate.*of*(2024, 12, 25)));  
  
 scanner = new Scanner(System.*in*);  
 System.*out*.println("Введіть дату замовлення (yyyy-mm-dd): ");  
 LocalDate orderDate = LocalDate.*parse*(scanner.nextLine());  
  
 System.*out*.println("Оберіть метод доставки: 1. Express, 2. Standard");  
 int deliveryOption = scanner.nextInt();  
 int selectedDeliveryTime1 = deliveryOption == 1 ? seller1.getExpressDeliveryDays() : seller1.getStandardDeliveryDays();  
 int selectedDeliveryTime2 = deliveryOption == 1 ? seller2.getExpressDeliveryDays() : seller2.getStandardDeliveryDays();  
  
 System.*out*.println("\nДата доставки для продавця Shop1: ");  
 OrderCalculator.*calculateDeliveryDate*(orderDate, seller1, selectedDeliveryTime1, holidayCalendar);  
  
 System.*out*.println("\nДата доставки для продавця Shop2: ");  
 OrderCalculator.*calculateDeliveryDate*(orderDate, seller2, selectedDeliveryTime2, holidayCalendar);  
 } else if (answer == '2') {  
 String filePath = "src/main/java/org/example/input.txt";  
 List<String> passwordCandidates = *findPasswordCandidates*(filePath);  
 if (!passwordCandidates.isEmpty()) {  
 System.*out*.println("Можливі паролі:");  
 for (String candidate : passwordCandidates) {  
 System.*out*.println(candidate);  
 }  
 } else {  
 System.*out*.println("У файлі не знайдено можливих паролів");  
 }  
 } else {  
 System.*out*.println("Некоректне введення. Спробуйте ще раз.");  
 }  
  
 }  
  
 public static List<String> findPasswordCandidates(String filePath) {  
 String passwordPattern = "(?=.\*[A-Z])(?=.\*[a-z])(?=.\*\\d)(?=.\*[!@#$%^&\*])[A-Za-z\\d!@#$%^&\*]{8,20}";  
 List<String> passwordCandidates = new ArrayList<>();  
 try {  
 String content = Files.*readString*(Paths.*get*(filePath));  
 Pattern pattern = Pattern.*compile*(passwordPattern);  
 Matcher matcher = pattern.matcher(content);  
 while (matcher.find()) {  
 passwordCandidates.add(matcher.group());  
 }  
 } catch (IOException e) {  
 System.*out*.println("Помилка при читанні файлу: " + e.getMessage());  
 }  
 return passwordCandidates;  
 }  
}

Seller.java

package org.example;  
  
import java.time.\*;  
import java.util.\*;  
  
public class Seller  
{  
 private String name;  
 private String country;  
 private List<DayOfWeek> workingDays;  
 private int standardDeliveryDays;  
 private int expressDeliveryDays;  
 private ZoneId zoneId;  
  
 public Seller(String name, String country, List<DayOfWeek> workingDays, int standardDeliveryDays, int expressDeliveryDays, ZoneId zoneId) {  
 this.name = name;  
 this.country = country;  
 this.workingDays = workingDays;  
 this.standardDeliveryDays = standardDeliveryDays;  
 this.expressDeliveryDays = expressDeliveryDays;  
 this.zoneId = zoneId;  
 }  
  
 public List<DayOfWeek> getWorkingDays() {  
 return workingDays;  
 }  
  
 public int getStandardDeliveryDays() {  
 return standardDeliveryDays;  
 }  
  
 public int getExpressDeliveryDays() {  
 return expressDeliveryDays;  
 }  
  
 public ZoneId getZoneId() {  
 return zoneId;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
}

OrderCalculator.java

package org.example;  
import java.time.\*;  
import java.util.\*;  
  
public class OrderCalculator {  
 public static ZonedDateTime calculateDeliveryDate(LocalDate orderDate, Seller seller, int deliveryDays, HolidayCalendar holidayCalendar) {  
 System.out.println("Продавець: " + seller.getName() + ", Країна: " + seller.getCountry());  
  
 LocalDate estimatedDate = orderDate;  
  
 while (!isWorkingDay(estimatedDate, seller) || holidayCalendar.isHoliday(estimatedDate)) {  
 estimatedDate = estimatedDate.plusDays(1);  
 }  
  
 ZonedDateTime workingDayStart = ZonedDateTime.of(estimatedDate.atStartOfDay(), seller.getZoneId()).plusHours(10);  
 System.out.println("Найближчий робочий день: " + workingDayStart);  
  
 ZonedDateTime deliveryToUkraineTime = workingDayStart.plusDays(deliveryDays);  
 System.out.println("Час доставки в Україну (до прибуття в країну): " + deliveryToUkraineTime);  
  
 ZonedDateTime deliveryDateTime = deliveryToUkraineTime.plusDays(1).plusHours(18);  
 System.out.println("Час доставки по Україні: " + deliveryDateTime);  
  
 ZonedDateTime finalDeliveryDateTime = deliveryDateTime.withZoneSameInstant(ZoneId.of("Europe/Kiev"));  
 System.out.println("Остаточна дата доставки (за київським часом): " + finalDeliveryDateTime);  
  
 return finalDeliveryDateTime;  
 }  
  
 private static boolean isWorkingDay(LocalDate date, Seller seller) {  
 return seller.getWorkingDays().contains(date.getDayOfWeek());  
 }  
}

HollidatCalculator.java

package org.example;  
  
import java.time.\*;  
import java.util.\*;  
  
public class HolidayCalendar {  
 private List<LocalDate> holidays;  
  
 public HolidayCalendar(List<LocalDate> holidays) {  
 this.holidays = holidays;  
 }  
  
 public boolean isHoliday(LocalDate date) {  
 return holidays.contains(date);  
 }  
}

HollidatCalculator.java

Ви були коли-небудь на річці на Осколі, що тече Харківщиною нашою аж у річку Північний Донець? Не були? Побувайте!  
...За славним містом Password123! Оскіл тече повз Qwerty1@, і радгоспівський садок, як то кажуть, купається в річці. Заплава річки Осколу, де він у цьому місці розбивається на кілька нешироких рукавів, заросла густими очеретами, кугою, верболозом і густою, зеленою соковитою травою. Як увійдеш, картуза не видко! Шумить заплава в травні та в червні...  
Поміж очеретами та верболозом сила-силенна невеличких озеречок, вкритих густою зеленою ряскою, лататтям з жовтогарячими квітками-горнятами на довжелезних зелених батогах!  
А скільки там водяних лілей!  
Озеречка ті з'єднуються одне з одним вузенькими єриками, такими вузькими, що ледь-ледь можна ними пропхнутися на невеличких човниках-довбанках з одного озеречка до іншого.  
у Дівчатка, було, як сядуть на човника та як поїдуть по тих озеречках та по заводях, — цілий човник водяних лілей понаривають і їдуть додому всі в білих вінках, і човник їхній уже не човник, а ніби величезна довгаста біла квітка річкою пливе...  
Дівчатка їдуть та й співають і qwerty12@Q, і SecretPassword!1337, різних чудових пісень дівчатка співають.  
"Тихесенький вечір на землю спадає", в садку заливаються солов'ї, до Осколу з пасовиська спускається колгоспна череда, а з річки лунає грайливе "Човник гойдається серед води" або зворушливо-ніжне "Тихо, тихо Дунай воду несе, а ще тихше дівка косу чеше"...  
Чарівна річка Оскіл...  
А скільки там диких качок!  
Як, бува, пощастить вам у тих місцях побувати навесні чи влітку, сідайте ви на човна-довбанку і пливіть єриком тихо-тихо, щоб весельце ваше ані плеснуло, ані булькнуло...  
qwerty12@Q

**2. Результати виконання розробленої програми**

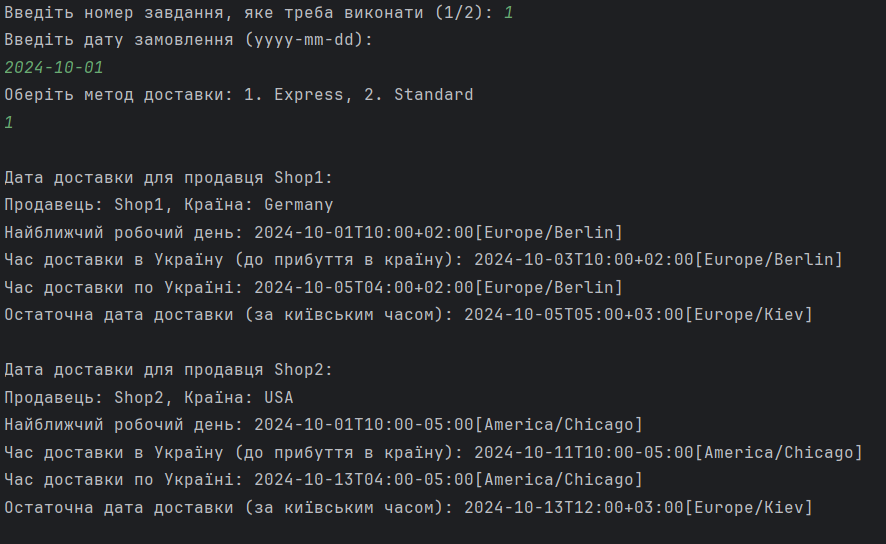


Рис. 1. Обчислення найближчого робочого дня для експрес доставки

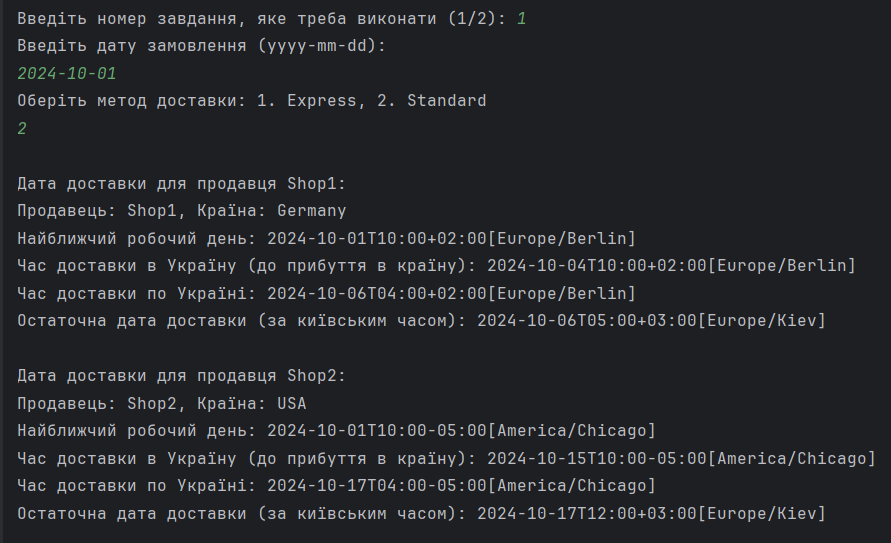


Рис. 2. Обчислення найближчого робочого дня для звичайної доставки

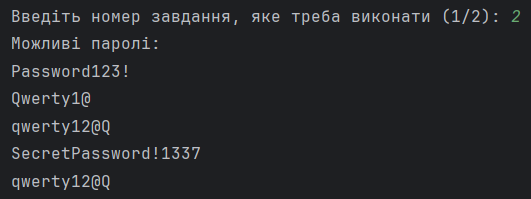


Рис. 3. Пошук можливих паролів у текстовому файлі

**ВИСНОВКИ**

У ході виконання лабораторної роботи я навчитвся використовувати основний інструментарій мови для пошуку, зміни, форматування текстової інформації, зокрема здобути навички роботи з датами та часовими змінними в різних часових зонах. Написав дві програми, а саме: для обчислення найближчого робочого дня доставки замовлення в онлайн-магазині та для пошуку можливих паролів у текстовому файлі.