# Національний технічний університет України

*Додаток 1*

# «Київський політехнічний інститут імені Ігоря Сікорського»

**Навчально-науковий інститут атомної і теплової енергетики**

**Кафедра цифрових технологій в енергетиці**

**ЗВІТ**

# з лабораторної роботи № 3

**з дисципліни «Програмування на мові Java»**

# Тема 3. ««Розробка програм в середовищі INTELLIJ IDEA для дослідження та особливостей забезпечення безпеки даних»

Варіант № 25

Виконав: Студент групи ТР-12

Руденко Владислав

Дата здачі: 29.10

Київ – 2023

**Завдання**

1. Написати програму мовою Java. Варіант обрати за списком групи. Представити виконання в Intellij IDEA. Продемонструвати детальні скріни виконання коду програми з поясненням. Обов’язково – наявність висновків.

2. Написати програму мовою Java. Варіант обрати за списком групи. Представити виконання в Intellij IDEA. Продемонструвати детальні скріни виконання коду програми з поясненням. Обов’язково – наявність висновків.

3. Зробити звіт з лабораторної роботи та вчасно надіслати викладачу на перевірку (дедлайн для надсилання звітів по Лаб\_3 – до 29.10.2023 року до 23:59).

**ЗАВДАННЯ 1**

**Варіант 1 (для всіх студентів, хто в списку йде за непарним номером (1,3,5,7 і т д))**

Написати програму мовою Java для обчислення часу, що необхідно для перебору всіх можливих варіантів паролів у системі із наступними параметрами: А=10, що складається з 7 елементів (L=7). Обрати неінтерактивну швидкість підбору паролів (V=10 паролів/сек). Вивести результат у вигляді текстового повідомлення, що покаже скільки годин, хвилин, секунд потрібно для виконання завдання (приклад виведеного повідомлення: Час перебору всіх паролів: ..годин,… хвилин.. секунд). Якщо час більше 24 годин – додати поле «дні». Обов’язково додати перевірку на валідність введеної інформації користувачем

**ЗАВДАННЯ 2**

**Варіант 1 (для всіх студентів, хто в списку йде за непарним номером (1,3,5,7 і т д))**

Написати програму мовою Java, яка приймає пароль від користувача та перевіряє його за вказаними критеріями безпеки. Пароль повинен відповідати наступним вимогам: - довжина не менше 8 символів; - містити як мінімум одну літеру верхнього регістру, одну літеру нижнього регістру та одну цифру. - не містити особисту інформацію користувача (ім'я, прізвище або дата народження). Обов’язково додати перевірку на валідність введеної інформації користувачем. При умові виконання всіх вимог, на екран виводиться повідомлення «Пароль надійний». При умові порушення хоча б однієї вимоги щодо безпеки – «Пароль не відповідає вимогам безпеки»

**ЗАВДАННЯ 3\***

*(завдання додаткове, для бажаючих отримати додаткові бали (оцінюється в 3 бали). Вимоги щодо оформлення такі ж як і для інших завдань)*

Написати програму мовою Java, яка приховує конфіденційні дані в текстовому повідомлені. Користувач вводить будь-яке текстове повідомлення (студент особисто обирає приклади текстових повідомлень: в першому повідомлені повинна бути вказана інформація з умовними даними платіжної картки (16 цифр, послідовно або через пробіл), в другому повідомлені – повинен з’явитися номер телефону (у форматі 0970000000 або 097-000-00-00), в третьому – комбінування двох попередніх варіантів повідомлень. При появі інформації про номер телефону чи реквізити платіжних карток, програма повинна приховати конфіденційні дані, замінивши їх на символ "\*" на кожен прихований символ. Наприклад, дані платіжної картки програма замінює на «\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*», а номер на «0\*\*\*\*\*\*\*\*\*» або «0\*\*-\*\*\*-\*\*-\*\*».

**Хід виконання та Результати**

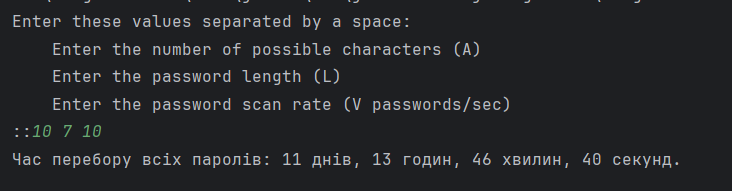


Рис. 1 Результат виконання Завдання 1

**Висновок:** В ході виконання завдання було створено програмний код для відображення часу перебору паролів. Відразу після введення базових параметрів програмно відбувається перевірка на валідність значень (вони мають бути більші 0) після чого за допомогою математичних функції відбувається обрахунок потужності паролів. Для пошуку часу по секундах, хвилинах, годинах, днях необхідно поділити попередню константу на кількість одиниць необхідного часу. Після обрахування для кожної змінної шукається відповідне значення модулю. За допомогою перевірки дні>0 можна визначити чи необхідно відобразити цю змінну. Це і дає нам змогу обрахувати необхідний час.

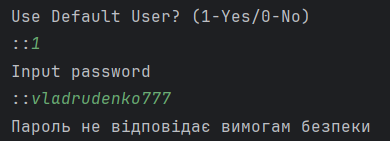


Рис. 2 Результат виконання Завдання 2

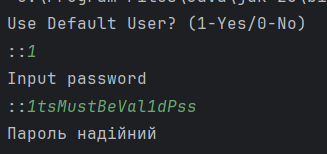


Рис. 3 Результат виконання Завдання 2 (Альтернативний варіант)

**Висновок:** В ході виконання другого завдання було створено валідатор паролю. Для збільшенні функціональності програмного коду було вирішено зробити об’єкт користувача (який зберігає данні Імені, Прізвища та дати народження) доступний для редагування, таким чином користувач може обрати між перевірки на основі існуючого користувача чи створити нового. Після зчитування паролю значення передається в деяку функцію (яка має повератємий тип boolean) де відбувається 2 дії: 1 – покрокова перевірка знаків на Верхній регістр, нижній регістр та число та 2 дія – перевірка вмісту інформації користувача в паролі. При позитивному результаті валідації користувачу виводиться повідомлення про успіх.

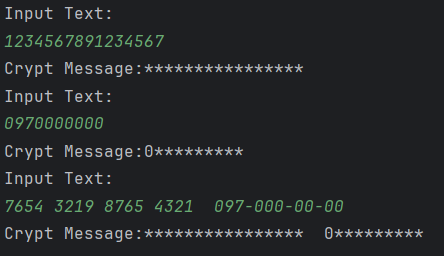


Рис. 4 Результат виконання Додаткового Завдання 3

**Висновок:** В ході роботи над додатковим завданням №3 було створено програму , яка призначена для обробки текстових повідомлень та захисту конфіденційних даних. Програма вміє виявляти конфіденційну інформацію, таку як номери платіжних карток та номери телефонів, введені користувачем у текстових повідомленнях. Потім вона застосовує регулярні вирази для заміни цих конфіденційних даних на символи "\*", щоб забезпечити конфіденційність даних.

*Додаток 1*

**Лістинг**

**Завдання 1**

package Lab3;

import java.util.Scanner;

public class Lab3Ex1 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter these values separated by a space:");

System.out.print("\tEnter the number of possible characters (А)\n\tEnter the password length (L)\n\tEnter the password scan rate (V passwords/sec)\n::");

int A = scan.nextInt();

int L = scan.nextInt();

int V = scan.nextInt();

if (L < 0 || V < 0 || A < 0) {

System.out.println("Error! Check input data.");

return;

}

long S = (long) Math.pow(A, L);

long seconds = S / V;

long minutes = seconds / 60;

long hours = minutes / 60;

long days = hours / 24;

seconds %= 60;

minutes %= 60;

hours %= 24;

System.out.print("Час перебору всіх паролів: ");

if (days > 0) {

System.out.print(days + " днів, ");

}

System.out.printf("%d годин, %d хвилин, %d секунд.%n", hours, minutes, seconds);

}

}

**Завдання 2**

package Lab3;

import java.util.Scanner;

public class Lab3Ex2 {

public static class User

{

private String FirstName;

private String SecondName;

private String BirthDate;

public User(String FirstName,String SecondName,String BirthDate )

{

this.FirstName = FirstName;

this.SecondName = SecondName;

this.BirthDate = BirthDate;

}

public boolean ValidCheck(String Password)

{

return !Password.toLowerCase().contains(FirstName.toLowerCase()) && !Password.toLowerCase().contains(SecondName.toLowerCase()) && !Password.toLowerCase().contains(BirthDate.toLowerCase());

}

}

public static boolean PasswordValid(String password, User account)

{

if(password.length() < 8)

{

return false;

}

boolean UpperValid = false,LowerValid= false,IntValid= false, UserCheck = false;

for (char letter : password.toCharArray())

{

if(Character.isUpperCase(letter))

UpperValid = true;

else if(Character.isLowerCase(letter))

LowerValid = true;

else if(Character.isDigit(letter))

IntValid = true;

}

if(account.ValidCheck(password))

{

UserCheck = true;

}

return UpperValid && LowerValid && IntValid && UserCheck;

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

User currentAccount = new User("Vlad","Rudenko","15-04-2004");

System.out.print("Use Default User? (1-Yes/0-No)\n::");

if(scan.nextInt() == 0)

{

scan.nextLine();

System.out.print("Input User First Name\n::");

String FName = scan.nextLine();

System.out.print("Input User Second Name\n::");

String SName = scan.nextLine();

System.out.print("Input User Birth Date (DD/MM/YYYY)\n::");

String Date = scan.nextLine();

currentAccount = new User(FName,SName,Date);

}

else

{

scan.nextLine();

}

System.out.print("Input password\n::");

String password = scan.nextLine();

if(PasswordValid(password, currentAccount))

{

System.out.println("Пароль надійний");

}

else

{

System.out.println("Пароль не відповідає вимогам безпеки");

}

}

}

**Завдання 3**

package Lab3;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Lab3Ex3 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

for(int i=0;i<3;i++)

{

System.out.println("Input Text:");

String message = scan.nextLine();

Matcher cardNumber = Pattern.compile("\\b\\d{4}(?:[ -]?\\d{4}){3}\\b").matcher(message);

message = cardNumber.replaceAll("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Matcher phoneNumberMatcher = Pattern.compile("\\b(0\\d{9}|0\\d{2}-\\d{3}-\\d{2}-\\d{2})\\b").matcher(message);

message = phoneNumberMatcher.replaceAll("0\*\*\*\*\*\*\*\*\*");

System.out.println("Crypt Message:" + message);

}

}

}