

## Лабораторна робота №2

### РОЗРОБКА КОНСОЛЬНИХ JAVA-ДОДАТКІВ

#### Мета роботи:

В даній лабораторній роботі розробляється консольний додаток для реалізації найпростішого додатка з використанням масивів, рядків та файлів.

#### Вказівки до роботи:

Консольна програма Java є відкомпільованим класом, що містить точку входу.

Розглянемо простий приклад:

```
public class First {  
    public static void main(String[] args) {  
        System.out.println("Перша програма на Java!");  
    }  
}
```

Тут клас First використовується тільки для того, щоб визначити метод main(), який є точкою входу і з якого починається виконання програми інтерпретатором Java. Метод main() містить аргументи-параметри командного рядка String[] args як масиву рядків і є відкритим (public) членом класу. Це означає, що метод main() видно і доступний будь-якому класу. Ключове слово static оголошує методи та змінні класу, що використовуються для роботи з класом загалом, а не лише з об'єктом класу. Символи верхнього та нижнього регістрів у Java різняться, чим Java нагадує мови C/C++ і PHP.

Виведення рядка "Перша програма на Java!" у прикладі здійснює метод println() (ln – перехід до нового рядка після виведення) властивості out класу System, який доступний в програмі автоматично разом із пакетом java.lang. Наведену програму необхідно помістити у файл, ім'я якого збігається з ім'ям класу та розширенням

java. Найпростіший спосіб компіляції написаної програми – виклик малого компілятора:

**javac First.java**

У разі успішної компіляції створюється файл First.class. Цей файл можна запустити на виконання з командного рядка за допомогою інтерпретатора Java наступним чином:

java First

*Для розробки програми можливе використання спеціальних засобів розробника.*

- **NetBeans IDE** – вільне інтегроване середовище розробки для всіх платформ Java – Java ME, Java SE та Java EE. Пропагується Sun Microsystems, розробником Java як базовий засіб для розробки ПЗ на мові Java.

- **Eclipse IDE** – вільне інтегроване середовище розробки Java SE, Java EE і Java ME. Пропагується IBM, одним з найважливіших розробників корпоративного програмного забезпечення, як базовий засіб для розробки програмного забезпечення на мові Java.

- **IntelliJ IDEA** – середовище розробки для платформ Java SE, Java EE та Java ME.

Розробник – компанія JetBrains. Поширюється у двох версіях: вільної безкоштовної (Community Edition) та комерційної пропрієтарної (Ultimate Edition).

- **JDeveloper** – середовище розробки для платформ Java SE, Java EE та Java ME.

Розробник - компанія Oracle.

- **JBuilder** -професійне інтегроване середовище розробки (IDE) на мові Java, засноване на програмному середовищі з відкритими кодами Eclipse.

- **BlueJ**– Середовище розробки програмного забезпечення мовою Java, створена в основному для використання у навчанні, але також підходяща для розробки невеликих програм.

Нижче розглянуто основні класи, що використовуються під час виконання лабораторної роботи, розглянуто приклад вирішення одного із завдань.

Клас java.io.File

Для роботи з файлами в програмах Java можуть бути використані класи з пакета java.io, одним із яких є клас File.

Клас File служить для зберігання та обробки як об'єкти каталогів та імен файлів. Цей клас не визначає способи роботи з вмістом файлу, але дозволяє маніпулювати такими властивостями файлу, як права доступу, дата і час створення, шлях до ієрархії каталогів, створення, видалення, зміна імені файлу та каталогу і т.д.

Основні методи класу File та способи їх застосування розглянуті в наступному прикладі:

```
import java.io.*;
import java.util.*;
public static void main(String[] args) throws IOException
/*відмова від обробки винятку в main()*/
{
```

```

    //3 об'єктом типу File асоціюється файл на диску. Способи створення об'єкта
    (прибрати "/" по черзі)
    //File fp = new File("com\learn\FileTest.java"); //File
    fp = new File("\com\learn", "FileTest.java"); //File fp
    = new File("d:\temp\demo.txt");
    File fp= new File("demo.txt");

    if(fp.isFile()) {//якщо об'єкт дисковий файл
        System.out.println("Ім'я файлу:\t" +fp.getName());
        System.out.println("Шлях до файлу:\t" +fp.getPath());
        System.out.println("Абсолютний шлях: t" +fp.getAbsolutePath());
        System.out.println("Канонічний шлях: t" +fp.getCanonicalPath());
        System.out.println("Розмір файлу:\t" +fp.length());
        System.out.println("Остання модифікація файлу:\t" +fp.lastModified());
        System.out.println("Файл доступний для читання:\t" +fp.canRead());
        System.out.println("Файл доступний для запису:\t" +fp.canWrite());
        System.out.println("Файл видалено:\t" +fp.delete());
    }

    if(fp.createNewFile()) {
        System.out.println("Файл " +fp.getName() + " створено");
    }

    if(fp.exists()) {
        System.out.println("temp файл "+fp.getName() + " існує");
    }

    else
        System.out.println("temp файл " +fp.getName() + " не існує"); //в
    об'єкт типу File міститься каталог\директорія
    File dir= newFile( "com\learn" );

    if(dir.isDirectory())/*якщо об'єкт оголошено як каталог на диску*/
        System.out.println("Директорія!");

    if(dir.exists()) {//якщо каталог існує
        System.out.println( "Dir" +dir.getName() + " існує " );
        File[]files=dir.listFiles();
        System.out.println("");

        for(int i=0; i < files.length; i++) {
            Date date= newDate(files[i].lastModified());
            System.out.println(files[i].getPath() + "\t|" +files[i].length() +
"\t|" + date.());
            //toLocaleString();//toGMTString()
        }
    }
}

```

У каталога (директорії) як об'єкту класу File є додаткова властивість – перегляд списку імен файлів за допомогою методів list(), listFiles(), listRoots().

## Клас System

Клас System містить набір корисних статичних методів та полів системного рівня. Примірник цього класу не може бути створений або отриманий.

Найбільш широко використовуваною можливістю, що надається System є стандартне виведення, доступне через змінну System.out. Стандартне виведення можна перенаправити на інший потік (файл, масив байт і т.д., головне, щоб це був об'єкт PrintStream, дивись документацію JSDK: <http://docs.oracle.com/javase/6/docs/api/> ):

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.PrintStream;

public class Second {
    public static void main(String[] args) {
        System.out.println("Study Java");

        try{
            PrintStream print= new PrintStream(new FileOutputStream("d:\\file2.txt"));
            //місце на диску, куди зберігається файл
            System.setOut(print);
            System.out.println("Study well");
        }

        catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

При запуску цього коду на екран буде виведено лише:

Study Java.

І файл "d:\file2.txt" буде записано:

Study well.

Аналогічно можуть бути перенаправлені стандартне введення `System.in` – викликом `System.setIn(InputStream)` та потік виведення повідомлень про помилки `System.err` – виклик `System.setErr` (за замовчуванням усі потоки – `in`, `out`, `err` – працюють з консоллю програми).

### Клас String

Клас String містить основні методи роботи з рядками:

- `concat(String s)` або `+` - злиття рядків;
- `equals(Object ob)`, `equalsIgnoreCase(String s)` – порівняння рядків з урахуванням та без урахування регістру;
- `compareTo(String s)`, `compareToIgnoreCase (String s)` – лексикографічне порівняння рядків з урахуванням та без урахування регістру;
- `contentEquals(StringBuffer ob)` – порівняння рядка та вмісту об'єкта типу `StringBuffer`;
- `charAt(int n)` – вилучення з рядка символу із зазначеним номером (Нумерація починається з нуля);
- `substring(int n, int m)`- витяг з рядка підрядка довжини `mn`, починаючи з позиції `n`;
- `length()` - Визначення довжини рядка;
- `valueOf(об'єкт)` – перетворення примітивного об'єкта до рядка;

- toUpperCase() / toLowerCase() – перетворення всіх символів рядка, що викликає, у верхній/нижній регістр;
- replace(char c1, char c2) – заміна у рядку всіх входжень першого символу другим символом;
- getBytes (параметри), getChars (параметри) - Витяг символів рядка у вигляді масиву байт або символів.

На наступному прикладі масив символів і ціле число перетворюються на об'єкти типу String з використанням методів цього класу.

```
public class DemoString {
    public static void main(String[] args) {
        chars[] = { 'J', 'a', 'v', 'a' };
        int i = 2;
        // коментар містить результат виконання коду
        String str = new String(s);
        // str = "Java"
        i = str.length(); // i = 4
        String num = String.(2);
        // num = "2"
        str = str.toUpperCase();
        // str = "Java"
        num = str.concat(num);
        // num = "Java2"
        str = str + "C"; // str = "JavaC";
        char ch = str.charAt(2);
        // ch = 'v'
        i = str.lastIndexOf( 'A' );
        // i = 3 (-1 якщо відсутня )
        num = num.replace('2', 'H'); // num = "JavaH"
        i = num.compareTo(str);
        // i = 5 ( між символами 'H' і '3' )
        str.substring(0, 3).toLowerCase(); // java
    }
}
```

### Приклад консольного java-додатку:

**Завдання:** Введіть n рядків з консолі. Вивести на консоль рядки та їх довжини, упорядковані за зростанням.

*Рішення:*

```
import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int n = 0;
        while(true) // введення числа рядків
        {
            System.out.println("Введіть число рядків");
            Scanner sc1 = new Scanner(System.in);
            try{
                n = sc1.nextInt();
                break;
            }
            catch(InputMismatchException fg) {
                // якщо введене значення не є числом
                System.out.print("Ви запровадили не число. " );
            }
        }
    }
}
```

```

// створення масиву рядків
String[]str= newString[n];
Scanner sc2= newScanner(System.in);

for(int i = 0; i < n; i++) {
    System.out.println( " Введіть рядок № + (i+1));
    str[i] =sc2.nextLine();
}
//Сортування масиву рядків за довжиною
for( int i = 0; i < str. length -1; i++) {

    for(int j = i+1; j < str.length; j++) {

        if(str[i].length()<str[j].length()) {
            String temp=str[i];
            str[i] =str[j];
            str[j] =temp;
        }
    }
}

intmaxlength = str[0].length();
System.out.println("Рядки в порядку зменшення
довжини:"); for( int i = 0; i < str. length ; i++) {
    System.out.print(str[i]);
    for(int j = 0; j < maxlength - str[i].length(); j++)
        System.out.print(" ");
    System.out.println( " її довжина = " +str[i].length());
}
}

```

## ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

1. Ввести n рядків з консолі, знайти найкоротший рядок. Вивести цей рядок і її довжину.
2. Введіть n рядків з консолі. Упорядкувати та вивести рядки у порядку зростання їх довжин, а також (другий пріоритет) значень цих їх довжин.
3. Введіть n рядків з консолі. Вивести на консоль ті рядки, довжина яких менша за середню, також їх довжини.
4. У кожному слові тексту k-ю букву замінити заданим символом. Якщо k більше довжини слова, коригування не виконувати.
5. У російському тексті кожну літеру замінити її номером у алфавіті. В одному рядку друкувати текст із двома пробілами між літерами, у наступному рядку внизу під кожною літерою друкуватиме її номер.
6. З невеликого тексту видалити всі символи, крім пробілів, які не є літерами. Між послідовностями поспіль літер, що йдуть, залишити хоча б одну прогалину.
7. З тексту видалити всі слова заданої довжини, що починаються на приголосну букву.
8. У тексті знайти всі пари слів, у тому числі одне є зверненням іншого.
9. Знайти та надрукувати, скільки разів повторюється у тексті кожне слово.

10. Знайти, яких літер, голосних чи приголосних, більше у кожному реченні тексту.
11. Вибрати три різні точки заданої на площині безлічі точок, складові трикутник найбільшого периметра
12. Знайти таку точку заданої на площині безлічі точок, сума відстаней від якої до решти мінімальна.
13. Опуклий багатокутник заданий на площині перерахуванням координат вершин у порядку обходу межі. Визначити площу багатокутника.