

№3 Тема: Утилітарні класи. Обробка масивів і рядків.

Мета: Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

1 Індивідуальне завдання

1.1 Розробник

Студент Малохвій Едуард Едуардович, КІТ-26А, Варіант 8 (Завдання №8).

1.2 Вимоги

- Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи (Class String, Manipulating Characters in a String, Comparing Strings and Portions of Strings).
- При вирішенні прикладних задач використовувати латинку.
- Продемонструвати використання об'єктів класу StringBuilder або StringBuffer.
- Для обробки даних використовувати класи-утиліти (особливий випадок допоміжного класу, див. Helper Class).
- Забороняється використовувати засоби обробки регулярних виразів.

1.3 Завдання

Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом навпаки) іншого (наприклад: "abc"-"cba", "def"-"fed"). Результат вивести у вигляді таблиці.

2 Розробка програми

2.1 Засоби ООП

Під час вирішенн поставленої задачі було використано ітератори, для отримання слів із колекції слів, таким чином змінюючи ітератор колекції можливо вдосконалювати алгоритм. Тобто, наприклад отримати анаграми лише для слів не

більших певного розміру, або певної літери на початку.

2.2 Ієрархія та структура класів

Проект містить наступні пакети:

- text - містить у собі необхідні класи, для обробки тексту на наявність анаграм.

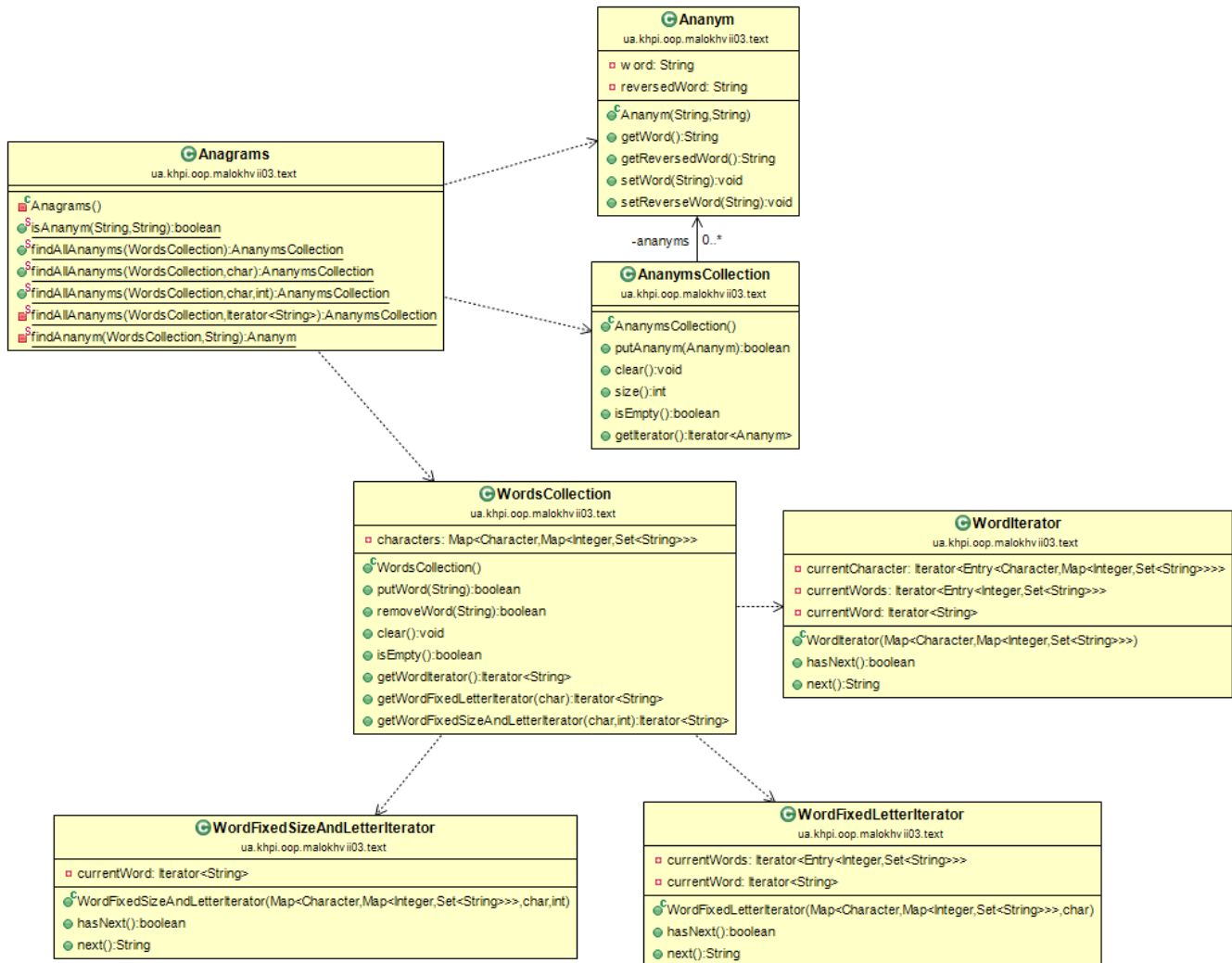


Рис. 1 - Діаграма класів із пакету text

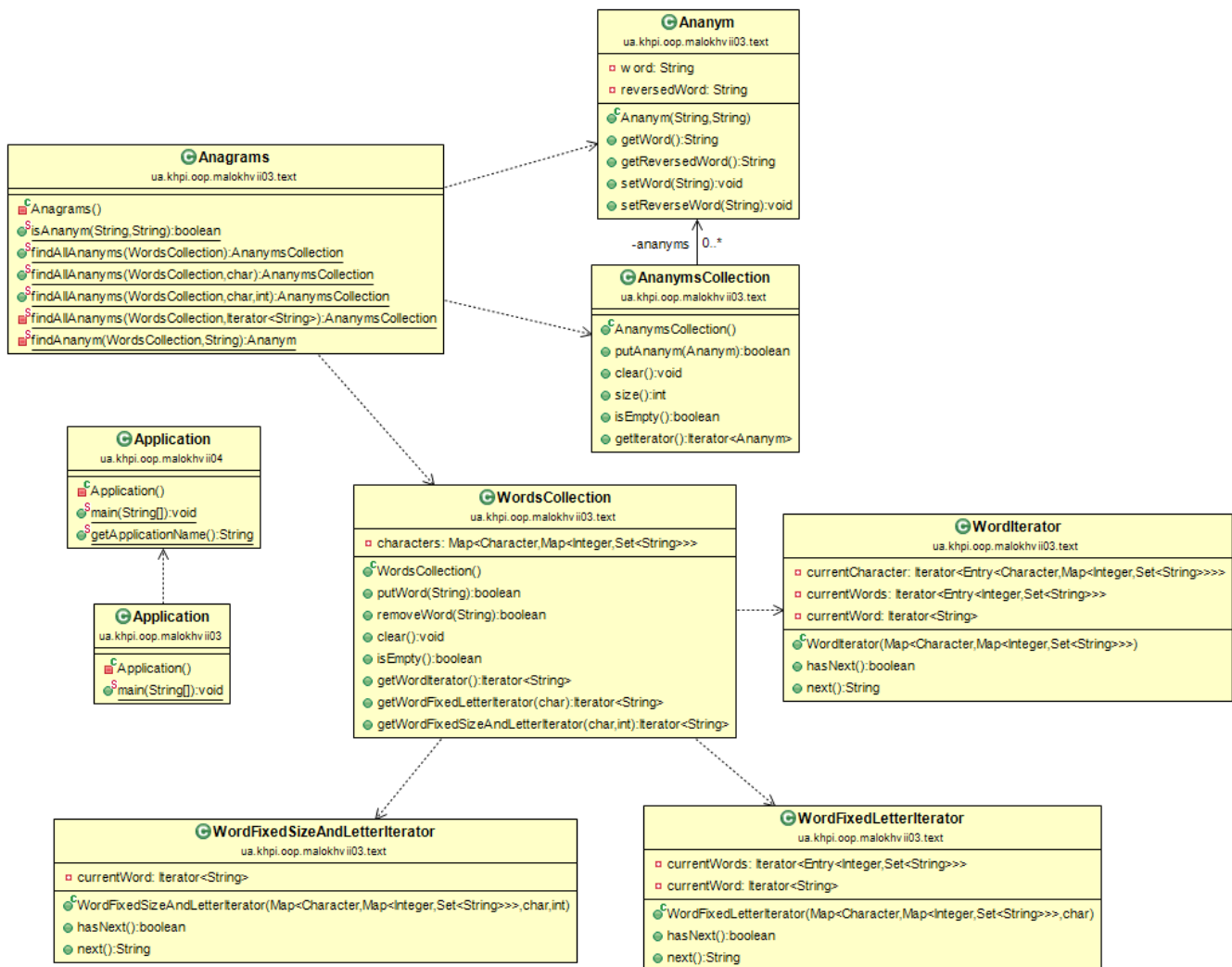


Рис. 2 - Загальна діаграма класів

2.3 Опис програми

Для надання інтерактивної оболонки використано рішення з пакету `ua.khpi.oop.malokhvii04`. Для індексування вхідних слів було використано декілька хеш-таблиць та хеш-множину для збереження слів.

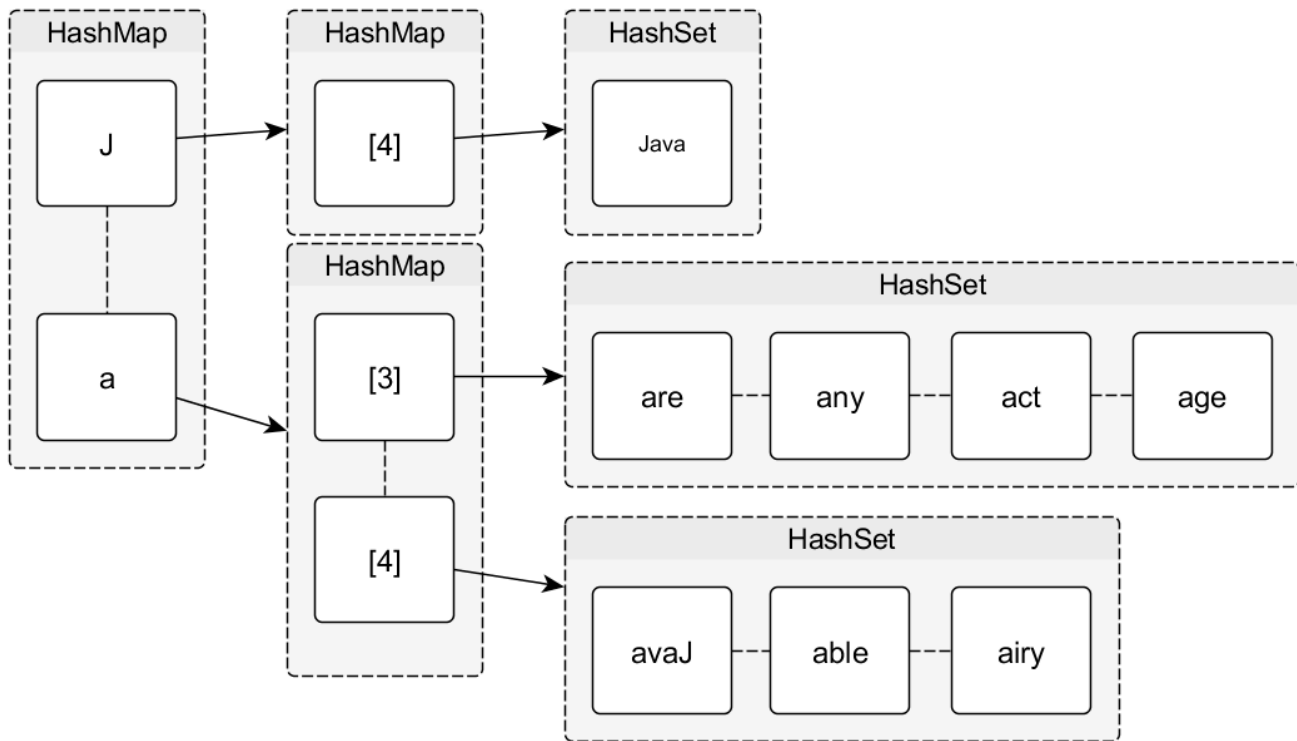


Рис. 3 - Приклад структури колекції слів

2.4 Важливі фрагменти програми

Нижче наведено фрагмент утилітарного класу для пошуку ананімів. Інші фрагменти детальніше див за посиланням (https://sourceforge.net/p/kit26a-cpp/code/HEAD/tree/malokhvii_eduard/src/ua/khpi/oop/malokhvii03/).

```

package ua.khpi.oop.malokhvii03.text;

import java.util.Iterator;

public final class Anagrams {

    private Anagrams() {

    }

    public static boolean isAnonym(final String word,
        final String reversedWord) {
        if (word.length() != reversedWord.length()) {
            return false;
        }

        char[] wordCharacterSequence = word.toCharArray();
        char[] reversedWordCharacterSequence = reversedWord.toCharArray();
  
```

```

    int charactersDifference = 0;
    for (int i = 0; i < wordCharacterSequence.length; i++) {
        charactersDifference += wordCharacterSequence[i]
            - reversedWordCharacterSequence[wordCharacterSequence.length
                - 1 - i];
        if (charactersDifference != 0) {
            return false;
        }
    }

    return true;
}

public static AnanymsCollection findAllAnanyms(
    final WordsCollection words) {
    return findAllAnanyms(words, words.getWordIterator());
}

public static AnanymsCollection findAllAnanyms(final WordsCollection words,
    final char letter) {
    return findAllAnanyms(words, words.getWordFixedLetterIterator(letter));
}

public static AnanymsCollection findAllAnanyms(final WordsCollection words,
    final char letter, final int size) {
    return findAllAnanyms(words,
        words.getWordFixedSizeAndLetterIterator(letter, size));
}

private static AnanymsCollection findAllAnanyms(final WordsCollection words,
    final Iterator<String> wordIterator) {
    AnanymsCollection ananyms = new AnanymsCollection();

    while (wordIterator.hasNext()) {
        Anonym anonym = findAnonym(words, wordIterator.next());
        if (anonym != null) {
            ananyms.putAnonym(anonym);
        }
    }

    return ananyms;
}

private static Anonym findAnonym(final WordsCollection words,
    final String word) {
    int wordLength = word.length();
    char lastLetter = word.charAt(wordLength - 1);

    Iterator<String> reversedWordIterator = words
        .getWordFixedSizeAndLetterIterator(lastLetter, wordLength);
    while (reversedWordIterator.hasNext()) {
        String reversedWord = reversedWordIterator.next();
        if (isAnonym(word, reversedWord)) {
            return new Anonym(word, reversedWord);
        }
    }

    return null;
}

```

```
}  
}
```

3 Результати роботи

Нижче наведено виведення обчислень у вигляді інтерактивної консолі.

```
Ed@malokhvii-ee MINGW32 /d/home/projects/ua.khpi.oop.malokhvii  
$ java -jar ananys.jar  
Shell was launched  
... Description: Used to find in the text ananys, such as "def" - "fed"  
... An ananym is a word whose spelling is derived by reversing the spelling  
... of another word. It is therefore a special type of anagram  
  
>>> -h  
  
Usage: ananys.jar  
... -debug -d          Switch on debug information  
... -exit -e           End interactive console session  
... -help -h           Get help on all the interactive console commands  
... -history -hs       Show commands call history  
... -input -i          Input of input data in the form of text files  
... -no-debug -nd      Switch off debug information  
... -output -o         Output of an anagram search result in the input text  
... -process -p        Search anagrams in the indexed input text  
  
>>> -i  
  
Please, enter file path with input text:  
... D:\home\projects\ua.khpi.oop.malokhvii\src\ua\khpi\oop\malokhvii03\text-259.txt  
  
>>> -p  
  
>>> -o  
  
List of words loaded from text file:  
... [A]: As  
... [C]: Compiler Classpath Community Corporation  
... [G]: GNU General Gosling  
... [I]: It IcedTea  
... [J]: JVM Java James  
... [L]: License  
... [M]: May Microsystems  
... [O]: Others Oracle
```

Рис. 4 - Фрагмент демонстраційної програми

```
>>> -o  
  
List of words loaded from text file:  
... [A]: As  
... [C]: Compiler Classpath Community Corporation  
... [G]: GNU General Gosling  
... [I]: It IcedTea  
... [J]: JVM Java James  
... [L]: License  
... [M]: May Microsystems  
... [O]: Others Oracle  
... [P]: Public Process  
... [S]: Sun  
... [T]: The  
... [W]: Web WORA  
... [a]: as at all are and any ava] also applets anywhere acquired  
... application alternative applications architecture  
... [b]: by but bew been based browser bytecode  
... [c]: can core code class computer compiled compiler component compilers compliance concurrent  
...  
... [d]: dna derives detroper designed detroppus developed developers dependencies  
... [e]: eht ehT era esu eno eerf either elcar0  
... [f]: fo for few from fewer facilities  
... [g]: general gnimargorp  
... [h]: has have htiw htob hcihw hguohtla  
... [i]: in is it its intended implementation implementations  
... [l]: low let level licenses language libraries laicremmoc  
... [m]: most much meaning machine machines  
... [n]: no ni need noillim noisrev  
... [o]: of on once object original oriented originally  
... [p]: plugin purpose possible platform platforms proprietary programming  
... [r]: rof run rehto revres reilrae ralupop released reference relicensed regardless recompilation  
...  
... [s]: si sA such sisab since syntax support standard snoisrev segaungal seinapmoc  
... srepolaved snoitacilppa specifically specifications  
... [t]: to the that than them tsom these tneilc tsetal typically technologies  
...  
... [u]: under
```

Рис. 5 - Фрагмент демонстраційної програми

```

Result of searching ananymys in text:
... Amount of ananymys: 39

Table of ananymys from text:
... -----
... Word                               Reversed word
... -----
... As                                 sA
... Java                              avaJ
... Oracle                            elcar0
... The                               eht
... are                               era
... and                               dna
... avaJ                              Java
... applications                     snoitacilppa
... by                                yb
... dna                               and
... developers                       srepoleved
... eht                              the
... ehT                              The
... era                              are
... elcar0                           Oracle
... fo                               of
... for                              rof
... gnimmargorp                      programming
... htiw                             with
... hcihw                            which
... in                               ni
... is                               si
... level                            level
... most                             tsom
... no                               on
... ni                               in
... of                               fo
... on                               no
... programming                     gnimmargorp
... rof                              for

```

Рис. 6 - Фрагмент демонстраційної програми

```

>>> -d

Switch on debug infomation

>>> -hs

[Debug]: 23d337f9-4b42-4673-adf3-f684378fabe4
    at ua.khpi.oop.malokhvii04.shell.CommandWithDebug.execute(CommandWithDebug.java:72)
    at ua.khpi.oop.malokhvii04.shell.Shell.handleCommand(Shell.java:55)
    at ua.khpi.oop.malokhvii04.Application.main(Application.java:46)

... Package name: ua.khpi.oop.malokhvii04.shell
... Package vendor: null
... Package version: null

... Simple name: HistoryCommand
... Canonical name: ua.khpi.oop.malokhvii04.shell.HistoryCommand
... Class loader: sun.misc.Launcher$AppClassLoader

... Start execution HistoryCommand

... Index: 4
... Name: HistoryCommand
... Id: 23d337f9-4b42-4673-adf3-f684378fabe4

... Index: 3
... Name: DebugCommand
... Id: ee78e063-b6c7-42d0-a3c1-a3a5108b2d73

... Index: 2
... Name: ProcessCommand
... Id: 24563aba-94a6-49de-9d46-7f633c46bc88

... Index: 1
... Name: InputCommand
... Id: bc2ff1b3-9845-4f5b-b2a2-c3e0afc76c25

... Finish execution HistoryCommand

```

Рис. 7 - Фрагмент демонстраційної програми

Висновки

У ході виконання лабораторної роботи були покращені навички використання масивів, досліджено поведінку рядкового типу в мові Java. Вивчено принцип написання утилітарних класів.