**[Утилітарні класи. Обробка масивів і рядків](https://oop-khpi.gitlab.io/" \l "tasks)**

**Мета роботи**:

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

**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.
* При вирішенні прикладних задач використовувати [латинку](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%B0%D0%B1%D0%B5%D1%82%D0%BA%D0%B0).
* Продемонструвати використання об'єктів класу [StringBuilder](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html) або [StringBuffer](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuffer.html).
* Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. [Helper Class](https://en.wikipedia.org/wiki/Helper_class)) та для обробки даних використовувати відповідні [статичні](https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html) методи.
* Забороняється використовувати засоби обробки регулярних виразів: класи пакету [java.util.regex](https://docs.oracle.com/javase/8/docs/api/java/util/regex/package-summary.html) ([Pattern](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html), [Matcher](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html) та ін.), а також відповідні методи класу [String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) (matches, replace, replaceFirst, replaceAll, split).

**Розробник**: Наймитенко Сергій Ігорович КІТ-119а №15.

**Загальне завдання**:

Ввести текст. У тексті знайти та вивести всі слова-паліндроми (однаково читається в обох напрямках - зліва направо та справа наліво. Наприклад: "noon", "civic", "radar", "level", "rotor", "refer").

**Опис програми**

**Засоби ООП**: клас, метод.

**Структура класів:** один публічний клас Main з двома методами.

**Важливі фрагменти програми:**

Метод task:

void task(String str)

{

StringBuilder b\_str = new StringBuilder(str);

int size = b\_str.length();

char[] arr = new char[size];

b\_str.getChars(0, size,arr,0);

System.out.println(arr);

if(flag)

System.out.println("Пошук слів поліндромів");

for (int i=0,j= 0; j < size; j++)

{

if(j+1!=size)

{

if(j<size && (j==0||arr[j-1]==' ' || arr[j+1]==',' || arr[j-1]=='.' || arr[j-1]=='!' || arr[j-1]=='?' || arr[j-1]==':' || arr[j-1]==';'))

{

if(flag)

System.out.println("Знайдено початок слова, переведення 'і' аргумента на початок слова. і = " + i);

i=j;

}

if(j!= size && (arr[j+1]==' ' || arr[j+1]==',' || arr[j+1]=='.' || arr[j+1]=='!' || arr[j+1]=='?' || arr[j+1]==':' || arr[j+1]==';'))

{

if(flag)

System.out.println("Знайдено кінець слова. j = " + j);

for(int k=i,f=j; f!=k && f>=i && k<=j ;f--,k++)

{

if(flag)

System.out.println("Порівняння літер слова і - " + arr[i] + " j - " + arr[j]);

if(arr[k]==arr[f]&&(f>=0||k<=size))

if(k+1==f-1||f-1==k||k+1==f)

{

if(flag)

System.out.println("Знайдено поліндром");

help.Show(b\_str.substring(i, j+1));

}

}

}

}

}

}

**Висновки**

В ході виконання лабораторної роботи була розроблена функція що визначає слова-паліндроми в тексті та виводить їх на екран. Були розроблені власні утилітарні класи, та були набуті навички з розробки програм з використанням масивів і рядків.